



UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE ELECTROTECNIA Y COMPUTACIÓN
RECINTO UNIVERSITARIO SIMON BOLIVAR

“Desarrollo de un prototipo de sistema de **realidad virtual** para la
caracterización de piezas automotrices”

Autores

Br. Luis Favio Rosales López

Br. Joseline Auxiliadora Silva Torrez

Tutor

Ing. Flor de María Valle Izaguirre

Asesor

Ing. Nestor Antonio Traña Obando

Managua, Nicaragua

Febrero 2017



Dedicatoria

Joseline Silva Torrez

A Dios, por la vida, por permitirme llegar a este momento la culminación de mi formación profesional como ingeniera, por no dejarme caer ante las dificultades y por darme día a día las fuerzas necesarias y por cada una de sus bendiciones.

A mi Madre Claudia Torrez, por su amor incondicional, por creer en mí y apoyarme en todo momento, por enseñarme que por más difícil que sea el camino nunca se debe desfallecer. A Luis Favio Rosales, por tu apoyo, dedicación y entrega, porque a pesar de todo tuvimos las fuerzas para culminar nuestra formación profesional. Esto nunca hubiera sido posible sin ti. A la Ing. Flor de María Valle, que con su orientación y dedicación nos guio durante el desarrollo del presente trabajo monográfico.

Luis Favio Rosales

A Dios en primer lugar por darme el entendimiento y la fortaleza para culminar este trabajo con éxito y llegar a este momento tan importante de mi formación profesional.

A mi madre Martha Elena López Mendoza, por su amor y comprensión, por darme la mejor educación y enseñarme que hay que luchar y trabajar con dedicación para lograr los objetivos de la vida.

A mis abuelos por estar en todo momento brindándome sus sabios consejos. A Joseline Silva Torrez por tu apoyo incondicional y motivación constante, sin el equipo que formamos no habiéramos podido alcanzar esta meta.

A nuestra tutora Ing. Flor de María Valle por su disposición, dedicación y orientación en el desarrollo del presente trabajo.



Agradecimientos

Agradecemos a Dios por ser nuestro guía, nuestra luz y camino, y darnos la fortaleza para seguir adelante cada día.

A nuestros amigos que estuvieron a nuestro lado, apoyándonos con sus conocimientos y consejos.

A la empresa Repuestos San Agustín por todo el apoyo y facilidades que nos fueron otorgadas para llevar a cabo las actividades propuestas durante el desarrollo de esta tesis.

A la Universidad Nacional de Ingeniería y a los docentes que formaron parte de nuestra formación académica, por el conocimiento que nos transmitieron en el transcurso de la carrera y de manera especial y sincera nuestro reconocimiento y agradecimiento a nuestra tutora Ing. Flor de María Valle y a nuestro asesor Ing. Nestor Traña Obando; por su oportuna, precisa e instruida orientación para el logro del presente trabajo, además de brindarnos todo su apoyo y conocimiento.

Y a todas las personas que colaboraron para la finalización de este trabajo monográfico.



Resumen del tema

El presente trabajo monográfico titulado “Desarrollo de un prototipo de sistema de realidad virtual para la caracterización de piezas automotrices (SRVSA)” tiene como objetivo general desarrollar un prototipo de Sistema de realidad virtual no inmersivo para la empresa “Repuestos San Agustín” que permita la caracterización de piezas automotrices de los modelos de vehículos seleccionados del catálogo interno de la empresa.

El prototipo de sistema de realidad virtual incluye la búsqueda de piezas vehiculares en entorno visual realista. Para ello se hizo uso de:

- La herramienta de Mapeo objeto-relacional para la plataforma MVC, como lo es Entity Framework.
- La implementación de Bootstrap, un framework para desarrollar interfaces y diseños web adaptables basados en HTML5 y CSS3.
- El aprovechamiento de técnicas de diseños para objetos 3D
- Compaginación de objetos 3D con interfaces web a través de librerías de JavaScript

En el primer bloque del documento se detalla la naturaleza del trabajo de tesis mediante una breve introducción, la declaración de sus objetivos y la respectiva justificación.

El segundo bloque presenta el Marco Teórico con los sustentos metodológicos y tecnológicos aplicados en el desarrollo del prototipo.

El tercer bloque describe el Análisis y presentación de Resultado para cada una de las fases de la metodología de ingeniería de software para el desarrollo rápido de aplicaciones RAD (acrónimo en inglés de rapid application development).

Por último, el cuarto bloque presenta los principales logros y limitaciones durante el desarrollo de la tesis monográfica, las experiencias adquiridas, las recomendaciones y conclusiones finales.



Contenido

INTRODUCCIÓN	1
OBJETIVOS DEL PROYECTO MONOGRÁFICO	2
OBJETIVO GENERAL	2
OBJETIVOS ESPECÍFICOS	2
JUSTIFICACIÓN	3
MARCO TEÓRICO	4
REALIDAD VIRTUAL	4
<i>Características de la realidad virtual</i>	4
<i>Elementos de un sistema de Realidad Virtual</i>	4
<i>Clasificación de los sistemas de realidad virtual</i>	5
<i>Realidad virtual inmersiva</i>	5
<i>Realidad virtual no inmersiva</i>	6
TECNOLOGÍAS DE DESARROLLO PARA ENTORNOS VISUALES	6
<i>Herramienta de diseño 3D</i>	6
<i>Técnicas de modelado 3D</i>	7
<i>Lenguajes De Programación</i>	8
<i>Almacenamiento y Acceso a Datos</i>	14
METODOLOGÍA DE DESARROLLO	18
<i>RAD como metodología de desarrollo</i>	18
<i>Prototipos de sistema</i>	19
ANÁLISIS Y PRESENTACIÓN DE RESULTADOS	21
FASE DE PLANIFICACIÓN DE REQUERIMIENTOS	21
<i>Requerimientos funcionales</i>	21
<i>Requerimientos no funcionales</i>	22
<i>Estructura del Sistema (paquetes)</i>	23
<i>Calendarización (asignación de recursos)</i>	24
FASE DE DISEÑO DE USUARIO	25
<i>Modelado de gestión</i>	25
<i>Modelado de datos</i>	26
<i>Modelado de proceso</i>	48
<i>Modelo de navegación</i>	52
FASE DE CONSTRUCCIÓN	53
<i>Patrón de arquitectura de desarrollo MVC</i>	53
<i>Uso de MVC</i>	55
<i>Presentación de piezas modeladas en ambiente web</i>	60
FASE DE CIERRE	62
<i>Implementación de la aplicación</i>	62
<i>Pruebas del sistema</i>	63
<i>Primer prototipo de Serie SRVSA</i>	67
CONCLUSIONES Y RECOMENDACIONES	75
LIMITACIONES	75
EXPERIENCIAS Y LOGROS	75



CONCLUSIONES	76
RECOMENDACIONES	77
BIBLIOGRAFÍA WEB	78
ANEXOS	80
ANEXO 1: PLAN DE IMPLANTACIÓN	80
ANEXO 2: EVALUACIÓN PRESUPUESTAL	85

Índice de Ilustraciones

<i>Ilustración 1: Modelo de Realidad Virtual Cliente-Servidor</i>	<i>5</i>
<i>Ilustración 2: Fases de diseño RAD</i>	<i>18</i>
<i>Ilustración 3: Tipos de prototipo</i>	<i>19</i>
<i>Ilustración 4: Diagrama de paquetes del Sistema de Ventas Repuestos San Agustín.</i>	<i>23</i>
<i>Ilustración 5: Diagrama de paquetes.....</i>	<i>24</i>
<i>Ilustración 6: Interacción entre las Fases del RAD.....</i>	<i>24</i>
<i>Ilustración 7: Modelo de gestión.....</i>	<i>25</i>
<i>Ilustración 8: Paquete Administración</i>	<i>26</i>
<i>Ilustración 9: Imagen incorporada en software Modelador.....</i>	<i>27</i>
<i>Ilustración 10: Imagen diseñada en 3D con técnica NURBS</i>	<i>28</i>
<i>Ilustración 11: Diseño de pieza 3D.....</i>	<i>28</i>
<i>Ilustración 12: Paquete Repuesto</i>	<i>30</i>
<i>Ilustración 13: Paquete Cotización.....</i>	<i>31</i>
<i>Ilustración 14: Diagrama de Entidad-Relación SRVSA</i>	<i>32</i>
<i>Ilustración 15: Diagrama de Entidad-Relación Seguridad SRVSA.....</i>	<i>33</i>
<i>Ilustración 16: Diagrama de Entidad-Relación Configuración Reporte SRVSA</i>	<i>33</i>
<i>Ilustración 17: Diagrama de Entidad-Relación Cotización SRVSA</i>	<i>34</i>
<i>Ilustración 18: Diagrama de Actividad, ingreso al sistema</i>	<i>48</i>
<i>Ilustración 19: Diagrama de Actividad, administración de catálogos.....</i>	<i>49</i>
<i>Ilustración 20: Diagrama de Actividad, administración de catálogo valor.....</i>	<i>49</i>
<i>Ilustración 21: Diagrama de Actividad, administración de entrada</i>	<i>50</i>
<i>Ilustración 22: Diagrama de Actividad, Aplicar entrada.....</i>	<i>50</i>
<i>Ilustración 23: Diagrama de Actividad, Búsqueda de pieza por categoría usuario final</i>	<i>51</i>
<i>Ilustración 24: Diagrama de Actividad, Búsqueda de pieza por producto usuario final</i>	<i>51</i>
<i>Ilustración 25: Modelo de Navegación SRVSA</i>	<i>52</i>
<i>Ilustración 26: Modelo de Navegación de Administración SRVSA.....</i>	<i>52</i>
<i>Ilustración 27: Diagrama de paquetes SRVSA.....</i>	<i>53</i>
<i>Ilustración 28: Arquitectura MVC.....</i>	<i>54</i>
<i>Ilustración 29: Diagrama de Clase SRVSA.Models.....</i>	<i>55</i>
<i>Ilustración 30: Diagrama de Acceso a entidades de utilidades de SRVSA</i>	<i>56</i>
<i>Ilustración 31: Diagrama de Clase ReportUtil para configuración del visor de reportes.....</i>	<i>56</i>
<i>Ilustración 32: Diagrama de Acceso a entidades de seguridad de SRVSA</i>	<i>57</i>
<i>Ilustración 33: Diagrama de Controlador Home SRVSA</i>	<i>58</i>
<i>Ilustración 34: Diagrama de Controlador Partoem SRVSA</i>	<i>59</i>
<i>Ilustración 35: Diagrama de Vista Home SRVSA.....</i>	<i>59</i>



<i>Ilustración 36: Transferencia de código OEM Vista-Controlador.....</i>	<i>60</i>
<i>Ilustración 37: Vista parcial _DetallePartes.....</i>	<i>61</i>
<i>Ilustración 38: Llamado de modelo 3D.....</i>	<i>61</i>
<i>Ilustración 39: Incrustación de vista de modelo 3D en vista principal</i>	<i>62</i>
<i>Ilustración 40: Inicio de Sesión SRVSA</i>	<i>67</i>
<i>Ilustración 41: Pestaña Inicio Superior SRVSA.....</i>	<i>68</i>
<i>Ilustración 42: Pestaña Inicio Inferior SRVSA</i>	<i>68</i>
<i>Ilustración 43: Pestaña Partes OEM SRVSA</i>	<i>69</i>
<i>Ilustración 44: Pestaña Partes mecánicas SRVSA</i>	<i>69</i>
<i>Ilustración 45: Pestaña Administración Superior SRVSA.....</i>	<i>70</i>
<i>Ilustración 46: Pestaña Administración Inferior SRVSA</i>	<i>70</i>
<i>Ilustración 47: Interfaz de agregar vehículo</i>	<i>71</i>
<i>Ilustración 48: Interfaz de editar vehículo.....</i>	<i>71</i>
<i>Ilustración 49: Interfaz de agregar producto.....</i>	<i>72</i>
<i>Ilustración 50: Interfaz de editar producto</i>	<i>73</i>
<i>Ilustración 51: Interfaz de edición de cotización.....</i>	<i>73</i>
<i>Ilustración 52: Interfaz de Reporte de Cotización.....</i>	<i>74</i>
<i>Ilustración 53: Interfaz de Reporte de Cotizaciones por vendedor.....</i>	<i>74</i>

Índice de Tablas

<i>Tabla 1: Tabla de Requerimientos funcionales</i>	<i>22</i>
<i>Tabla 2: Tabla de base de datos usuario</i>	<i>35</i>
<i>Tabla 3: Tabla de base de datos rol</i>	<i>35</i>
<i>Tabla 4: Tabla de base de datos rolusuario</i>	<i>36</i>
<i>Tabla 5: Tabla de base de datos accion</i>	<i>36</i>
<i>Tabla 6: Tabla de base de datos accionxrol</i>	<i>37</i>
<i>Tabla 7: Tabla de base de datos catalogo</i>	<i>37</i>
<i>Tabla 8: Tabla de base de datos catalogovalor</i>	<i>37</i>
<i>Tabla 9: Tabla de base de datos directorio</i>	<i>38</i>
<i>Tabla 10: Tabla de base de datos vehiculo</i>	<i>39</i>
<i>Tabla 11: Tabla de base de datos producto</i>	<i>40</i>
<i>Tabla 12: Tabla de base de datos productovehiculo</i>	<i>40</i>
<i>Tabla 13: Tabla de base de datos precio</i>	<i>41</i>
<i>Tabla 14: Tabla de base de datos bodega</i>	<i>41</i>
<i>Tabla 15: Tabla de base de datos existencia</i>	<i>42</i>
<i>Tabla 16: Tabla de base de datos proveedor</i>	<i>43</i>
<i>Tabla 17: Tabla de base de datos entrada</i>	<i>43</i>
<i>Tabla 18: Tabla de base de datos entradadetalle</i>	<i>44</i>
<i>Tabla 19: Tabla de base de datos cliente</i>	<i>45</i>
<i>Tabla 20: Tabla de base de datos empleado</i>	<i>45</i>
<i>Tabla 21: Tabla de base de datos reporte</i>	<i>46</i>
<i>Tabla 22: Tabla de base de datos reporteConfiguracion</i>	<i>46</i>
<i>Tabla 23: Tabla de base de datos reporteCategoria</i>	<i>46</i>
<i>Tabla 24: Tabla de base de datos cotización</i>	<i>47</i>
<i>Tabla 25: Tabla de base de datos cotizaciondetalle</i>	<i>47</i>
<i>Tabla 26: Tabla de base de datos directoriovehiculo</i>	<i>48</i>
<i>Tabla 27: Carga de modelos 3D en sistema</i>	<i>63</i>
<i>Tabla 28: Manipulación de objetos 3D y contenido Web</i>	<i>63</i>
<i>Tabla 29: registrar un nuevo catalogo</i>	<i>64</i>



<i>Tabla 30: registrar un nuevo vehículo</i>	64
<i>Tabla 31: registrar nuevo producto</i>	64
<i>Tabla 32: registrar entrada</i>	65
<i>Tabla 33: registrar precio</i>	65
<i>Tabla 34: registrar proveedor</i>	66
<i>Tabla 35: registrar bodega</i>	66
<i>Tabla 36: Buscar pieza por categoría</i>	66
<i>Tabla 37: Solución a prueba 1</i>	66
<i>Tabla 38: Solución a prueba 2</i>	67



Introducción

El sistema de realidad virtual para la caracterización de piezas automotrices es una propuesta, basada en el prototipaje para identificar piezas automotrices comercializadas por Repuestos San Agustín, mediante un entorno visual realista.

La empresa “Repuestos San Agustín” está ubicada en la Ciudad de Boaco y se dedica a la comercialización de repuestos automotrices. Cuenta con un sistema administrativo llamado “Sistema de Ventas San Agustín”.

La mayoría de propietarios de vehículos desconocen la enorme cantidad de tecnología (componentes/mecanismos) que tienen alrededor. Es natural que la visión del chofer no sea la de valorar cómo se diseñan y fabrican los automóviles y sus piezas, sino evaluar su utilidad, ergonomía, estética y funcionalidad, por lo que se les hace difícil la identificación precisa de las piezas automotrices de sus vehículos.

Realidad virtual es aquel entorno informático que representa, de manera digital, algo que simula ser real. Esto quiere decir que la persona que usa la realidad virtual (o que “ingresa” en un entorno de este tipo) siente que está interactuando con elementos reales, aunque no lo sean en sentido físico.

La presente monografía propone el desarrollo de un Prototipo primero de serie (ver Prototipos de sistema pág. 19) de un Sistema de realidad virtual no inmersivo, que ayude a la empresa “Repuestos San Agustín” a brindar una mejor atención a sus clientes; al contar con una herramienta que les permita identificar la imagen de las piezas automotrices de los vehículos, su ubicación en el vehículo, así como otras características útiles de conocer para decidir sobre la adquisición o no de la pieza de refacción. Los vehículos seleccionados son de la marca Hyundai (Accent 2009) y KIA (Avella 1998, Rio 2013) y su elección es debido a la demanda recibida en el local.

Se pretende que este prototipo además de presentar las piezas de repuestos en 3D permita determinar la existencia de las piezas que busca el cliente en el almacén, además de generar una proforma para su posterior facturación. Este proyecto finalizará como un prototipo primero de serie para que posteriormente y a solicitud de la empresa Repuestos San Agustín se puedan incorporar nuevos modelos automotrices, a fin de complementar los catálogos vehiculares.



Objetivos del proyecto monográfico

Objetivo general

Desarrollar un prototipo de Sistema de realidad virtual no inmersivo para la empresa “Repuestos San Agustín” que permita la caracterización de piezas automotrices de los modelos de vehículos seleccionados del catálogo interno de la empresa.

Objetivos específicos

1. Desarrollar un prototipo de Sistema de realidad virtual que proporcione una modalidad alternativa para búsqueda de piezas de refacción automovilísticas a través de un entorno visual realista.
2. Implementar el módulo de proformas, con imágenes de las piezas a adquirir, para su cotización y posterior facturación.
3. Proporcionar una herramienta visual que sirva al personal de “Repuestos San Agustín” como método de aprendizaje visual en escala 3D para la caracterización de piezas automotrices.



Justificación

La industria automovilística se encarga del diseño, ensamblaje, reparación, venta de automóviles y autopartes. A pesar de la importancia que posee esta industria, el acceso a los catálogos de repuestos y manuales de reparación es limitado a ciertos sectores; los distribuidores de autopartes y algunas veces, talleres automotrices, se asisten de catálogos impresos con referencias vagas de las piezas.

En los últimos 2 años, hemos estado relacionados laboralmente con el sector automovilístico, tanto en el área de adquisiciones como en el área de ventas, lo que nos ha permitido ver de cerca la necesidad de reconocer con precisión las piezas automotrices.

Un caso particular es la tarea de verificar que las piezas que el proveedor ofertaba fueran 100% exactas a la pieza solicitada, para posteriormente hacer un pedido a gran escala de dicho producto. Al no contar con conocimientos automovilísticos ni muestras físicas, esta labor se volvía más complicada, ya que debía buscar en medios como internet y catálogos impresos la pieza que estaba verificando para compararla con la muestra ofertada por el proveedor. En caso de no encontrarla, debía recurrir a personas que tuvieran más conocimientos para identificarla correctamente. Si ninguna de las opciones para la identificación de la pieza funcionaba entonces no podría dar una respuesta adecuada a la necesidad, por lo cual la labor realizada era infructuosa, ocasionando retraso en la adquisición y potencialmente, la pérdida de las ventas asociadas.

Experiencias como éstas, ayudaron a identificar la oportunidad de proponer el sistema de realidad virtual para la caracterización de piezas automotrices.



Marco Teórico

Realidad virtual

Características de la realidad virtual

La Realidad Virtual es “una simulación generada por un computador en un ambiente de tres dimensiones, en donde el usuario puede visualizar y manipular el contenido de dicho ambiente” (Matsuba & Roehl, 1996).

De manera general, las características de la realidad virtual son:

- Se expresa en lenguaje gráfico tridimensional.
- Hace de 3D una herramienta dinámica e interactiva.
- Permite vivenciar experiencias controladas.
- Su comportamiento es dinámico y opera en tiempo real.
- Sus estímulos hacen real lo virtual.
- Su operación está basada en la incorporación del usuario en el “interior” del medio computarizado.
- Su relación con el usuario hace que el aprendizaje sea más intenso.
- Puede ser utilizada en toda la industria de la capacitación y entrenamiento.
- Abre las alternativas donde el único límite es la imaginación del hombre.

Elementos de un sistema de Realidad Virtual

- **Entrada de datos:** realiza un seguimiento del usuario y su interacción con el entorno virtual (guantes de datos, ratones 3D, etc.).
- **Salida de datos:** realimenta los dispositivos sensoriales del usuario: sonido, vídeo (HMD), tacto (guantes de datos).
- **Motor de Realidad Virtual:** máquina que alberga el software donde se creará el mundo virtual. Necesita un hardware acorde con la calidad de realidad virtual que se requiera. Normalmente se trata de estaciones gráficas de gran desempeño.
- **Software de Realidad Virtual:** lenguajes, librerías y sistemas autorizados que se usan para implementar interfaces completas para diferentes mundos virtuales.
- **Base de datos:** contiene los objetos del mundo virtual y sus propiedades.



El motor de Realidad Virtual puede estar basado en un modelo cliente-servidor de la siguiente forma:

Servidor:

- Sincronización de las imágenes.
- Control de colisión de las imágenes.
- Comunicación con los clientes, que visualizarán el mundo virtual.

Cliente:

- Contiene el mundo virtual.
- Se comunican con el servidor.

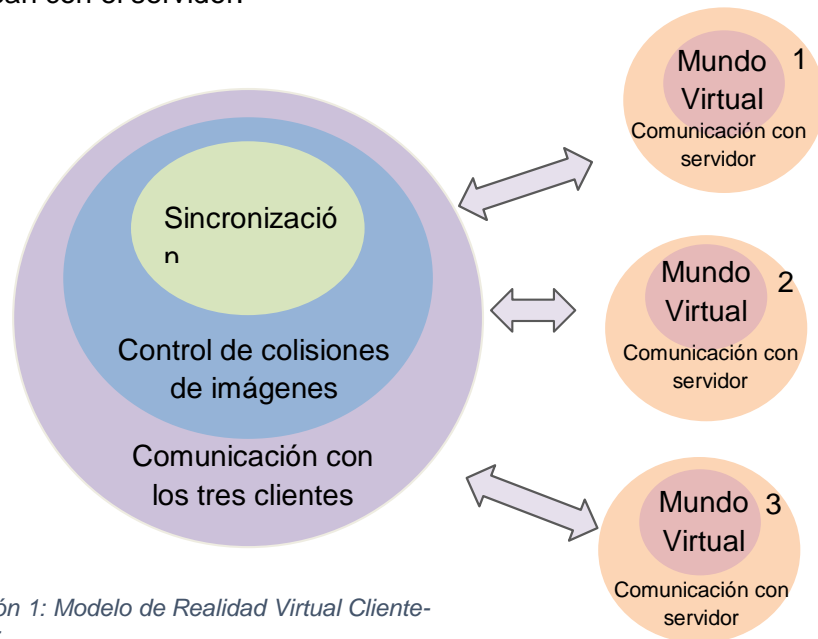


Ilustración 1: Modelo de Realidad Virtual Cliente-Servidor

Cada cliente tiene perspectivas distintas del mundo virtual, mientras que el servidor enviará órdenes de navegación para que los clientes actúen en consecuencia.

Clasificación de los sistemas de realidad virtual

La realidad virtual dependiendo del enfoque se puede dividir en 2 categorías, Realidad virtual inmersiva y realidad virtual no inmersiva, la diferencia entre estas es muy grande, pero ambas cumplen con el mismo propósito, de que el usuario se sienta dentro de un mundo que reacciona ante sus acciones.

Realidad virtual inmersiva

Consiste en una serie de dispositivos de tecnología más avanzada como suelen ser cascos, guantes u otros dispositivos que capturan la posición y rotación de diferentes



partes del cuerpo humano. Estos métodos inmersivos generalmente se relacionan con un ambiente tridimensional, ficticio, creado por computadora y en el cual uno se puede desplazar mediante los elementos ya mencionados anteriormente. Estos dispositivos son de alto costo y generalmente el usuario prefiere manipular el ambiente virtual por medio de dispositivos familiares como son el teclado y el ratón que por medio de cascos pesados o guantes.

Realidad virtual no inmersiva

La realidad virtual no inmersiva, da al usuario un ambiente en el cual puede interactuar en tiempo real, ya bien sea con un programa, con una persona que se encuentra en otro sitio o con un personaje ficticio que da órdenes y/o comparte con él, utiliza como único medio de comunicación el equipo de cómputo. En este caso la interacción se limita al uso de los dispositivos de hardware básicos como el teclado, mouse, monitor, tarjeta de sonido y bocinas.

La realidad virtual no inmersiva ofrece un nuevo mundo a través de una ventana de escritorio. Este enfoque tiene varias ventajas sobre el inmersivo, como el bajo costo, la facilidad y la rapidez de aceptación de los usuarios. Los dispositivos inmersivos son de elevado costo y generalmente el usuario prefiere manipular el ambiente virtual por medio de dispositivos familiares, por ejemplo, el teclado y el ratón, que a través de cascos pesados o guantes pesados. (Gálvez, 2004)

Tecnologías de desarrollo para entornos visuales

Herramienta de diseño 3D

CAD (Diseño asistido por computadora)

El diseño asistido por computadora es el uso de programas informáticos para crear representaciones gráficas de los objetos físicos en dos o tres dimensiones (2D o 3D).

CAD se utiliza en todo el proceso de ingeniería, desde el diseño conceptual y el diseño de productos, a través de la potencia y el análisis dinámico de los ensamblajes, hasta la



definición de los métodos de fabricación. Estas herramientas CAD/CAM/CAE como son Blender, Maya, Autodesk 3dsMax, entre otras; permiten el diseño de objetos 3D como también la conversión de planos 2D a 3D y gracias a sus opciones de renderizado permiten texturas realistas permitiendo que un ingeniero tanto de forma interactiva como automática analice variantes de diseño, para encontrar el diseño óptimo para la fabricación y reducir al mínimo el uso de prototipos físicos.

El diseño asistido por computadora ocupa un papel importante en la producción y el diseño del prototipo de un automóvil, CAD se utiliza principalmente para la creación de modelos 3D detallados, para ello se debe diseñar de forma exacta, clara y a escala real, con sus dimensiones, formas y características de cada pieza, auxiliándose de esquemas, diagramas, planos eléctricos y electrónicos, etc., para que ensamblen y funcionen de forma correcta, con el fin de no obtener margen de errores.

Técnicas de modelado 3D

Para la construcción de un modelo 3D, existen una serie de técnicas que pueden ser de mucha utilidad entre las que se encuentran:

1. Estructuras Predefinidas.
2. Box Modeling.
3. NURBS Modeling.
4. Operaciones Booleanas.
5. Extrude || Lathe.
6. Loft.
7. Sistema de Partículas.
8. Modelos por Texturas.

Cada una de estas técnicas presentan su particularidad y ventajas al momento de la representación de un modelo en 3D, en dependencia del resultado que se desea lograr se puede utilizar una o varias de esas técnicas de manera conjunto.

Para construir modelos con excelente definición se utilizan técnicas como NURBS las cuales proporcionan mallas de alta complejidad de aspecto orgánico o curvado que emplea como punto de partida splines (líneas simples) estas mallas se crean a partir de la anidación de varios splines.



La geometría NURBS tiene cinco cualidades esenciales que la convierten en la opción ideal para el modelado asistido por ordenador.

- Para el intercambio de la geometría NURBS, se utilizan distintos métodos estándar de la industria. Los usuarios pueden transportar todos sus modelos geométricos entre los diferentes programas de modelado, renderizado, animación e ingeniería de análisis que hay en el mercado. Estos programas pueden almacenar información geométrica que podrá ser utilizada en el futuro.
- Las NURBS tienen una definición precisa y muy conocida. La geometría NURBS se enseña en las facultades de matemáticas e informática de las universidades más importantes. Eso significa que los vendedores de software especializado, los equipos de ingenieros, las empresas de diseño industrial y las empresas de animación que necesitan crear aplicaciones de software específicas para sus proyectos podrán encontrar programadores capacitados para trabajar con la geometría NURBS.
- Las NURBS pueden representar con precisión objetos geométricos estándar tales como líneas, círculos, elipses, esferas y toroides, así como formas geométricas libres como carrocerías de coches y cuerpos humanos.
- La cantidad de información que requiere la representación de una forma geométrica en NURBS es muy inferior a la que necesitan por separado las aproximaciones comunes.
- La regla de cálculo de las NURBS, que se describe a continuación, se puede implementar en un ordenador de manera eficaz y precisa.

Lenguajes De Programación

Unity3D

Unity3D es un poderoso motor gráfico que es utilizado de manera comercial para el desarrollo de videojuegos. El editor es intuitivo y personalizado brindando mayor libertad en el trabajo. El sistema de UI permite crear interfaces de manera rápida e intuitiva. Unity3D es compatible con diversas plataformas como Android, iOS, Microsoft y permite crear video juegos para consolas como Xbox 360, PlayStation 3, PlayStation Vita.



Al ser multiplataforma cuenta con una ventajosa integración con Visual Studio C# permitiendo el desarrollo en este lenguaje de programación. (Technologies, 2014)

El Scripting de Unity3D es esencial, ya que define el comportamiento del sistema, es posible regular el comportamiento de objetos y eventos mediante la creación y asignación de variables.

JAVASCRIPT

JavaScript es un lenguaje de scripting multiplataforma y orientado a objetos. Es un lenguaje pequeño y liviano. Dentro de un ambiente de host, JavaScript puede conectarse a los objetos de su ambiente y proporcionar control programático sobre ellos. (Thomas Powel, 2004)

JavaScript contiene una librería estándar de objetos, tales como Array, Date, y Math, y un conjunto central de elementos del lenguaje, tales como operadores, estructuras de control, y sentencias. El núcleo de JavaScript puede extenderse para varios propósitos, complementándolo con objetos adicionales, por ejemplo:

Client-side JavaScript extiende el núcleo del lenguaje proporcionando objetos para controlar un navegador y su modelo de objetos (o DOM, por las iniciales de Document Object Model). Por ejemplo, las extensiones del lado del cliente permiten que una aplicación coloque elementos en un formulario HTML y responda a eventos del usuario, tales como clics del ratón, ingreso de datos al formulario y navegación de páginas. (Pollock, 2010)

Server-side JavaScript extiende el núcleo del lenguaje proporcionando objetos relevantes a la ejecución de JavaScript en un servidor. Por ejemplo, las extensiones del lado del servidor permiten que una aplicación se comuniquen con una base de datos, proporcionar continuidad de la información de una invocación de la aplicación a otra, o efectuar manipulación de archivos en un servidor.



Three.js

Es una biblioteca liviana escrita en JavaScript para crear y mostrar gráficos animados por ordenador en 3D en un navegador, aprovechando las grandes novedades que ofrece HTML5 para la generación de contenidos multimedia; aprovecha tanto las capacidades de HTML5 que es capaz de generar escenas 3D con WebGL, Canvas (2D) y SVG. (Autores, s.f.)

Características

- Renderizadores: Canvas, SVG y WebGL.
- Efectos: anáglifo, bizco y la barrera de paralaje.
- Escenas: añadir y eliminar objetos en tiempo de ejecución; niebla.
- Cámaras: perspectiva y ortográfica; controladores: trackball, FPS, trayectoria y otras.
- Animación: armaduras, cinemática directa, cinemática inversa, morphing y fotogramas clave.
- Luces: ambiente, dirección, luz de puntos y espacios, sombras: emite y recibe.
- Materiales: Lambert, Phong, sombreado suave, texturas y otras.
- Shaders: el acceso a las capacidades del OpenGL Shading Language (GLSL): reflejos en la lente, pase profundo y una extensa biblioteca de post-procesamiento
- Objetos: mallas, partículas, sprites, líneas, cintas, huesos y otros.
- Geometría: plano, cubo, esfera, toroide, texto en 3D y otras; modificadores: torno, extrusión y tubo.
- Cargadores de datos: binario, imagen, JSON y escena.
- Utilidades: conjunto completo de funciones matemáticas en 3D, incluyendo tronco, matriz Quaternion, UVs y otras.
- Exportación e importación: utilidades para crear archivos compatibles con JSON Three.js desde: Blender, openCTM, FBX, Max, y OBJ.
- Depuración: Stats.js, WebGL Inspector, Three.js Inspector.



Tecnología Web

Página Web

Es un documento electrónico el cual contiene información textual, visual y/o sonora que se encuentra alojado en un servidor y puede ser accesible mediante el uso de navegadores. Una página web forma parte de una colección de otras páginas webs dando lugar al denominado sitio web el cual se encuentra identificado bajo el nombre de un dominio.

La creación y desarrollo de una página web se realiza bajo un lenguaje de programación capaz de ser interpretados por los navegadores, lenguajes como el HTML, PHP, ASP, JSP o RUBY son ejemplos entre otros.

Cliente y servidor Web

El término “cliente” es usado en un entorno cliente – servidor para referirse al programa que utilizan las personas para ejecutar una aplicación. El cliente también puede ser un servidor que necesita hacer sus propias peticiones y que sólo pueden ser cumplidas por otros servidores (Leon Shklar, 2003). Uno de los clientes más utilizados, sobre todo por su versatilidad, es el navegador web.

El servidor es un programa que está esperando las solicitudes de conexión realizadas por los clientes web (Leon Shklar, 2003), generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web.

Un servidor web procesa los scripts del lado del servidor para dar una salida en HTML y otros lenguajes del lado del cliente al Navegador Web del cliente.

Aplicaciones Web

Se trata de una aplicación cliente servidor que utiliza un navegador como un programa cliente y lleva a cabo un servicio interactivo mediante la conexión con los servidores a través de internet (o intranet) (Leon Shklar, 2003).

Una aplicación web presenta contenido de forma dinámica, dependiendo de las solicitudes, los comportamientos de los usuarios y aspectos de seguridad.

Una ventaja clara del software basado en web es que todos los datos están centralizados y accesibles a través de la web desde cualquier plataforma en cualquier computadora y en cualquier momento.



HTTP

Es un protocolo utilizado para la transmisión de las páginas web. Este opera a través de peticiones y respuestas entre un cliente y un servidor. El navegador se comunica con el servidor de internet mediante este protocolo y se envían las paginas en el lenguaje html, el navegador lo interpreta y nos muestran su contenido.

HTTP (Leon Shklar, 2003) es un protocolo sin estado, es decir que no guarda ninguna información sobre conexiones anteriores, esto significa que establece la interacción entre el cliente y servidor para contener una secuencia de comandos. El servidor se requiere para mantener el "Estado" de la conexión a lo largo de la transmisión de comandos sucesivos, hasta que la conexión es terminada.

Ejemplo de Peticiones / Respuestas entre cliente y servidor:

El usuario escribe en su navegador web la URL de la página que desea visitar, determinada URL se traduce por el navegador en una petición HTTP. La estructura de los mensajes de solicitud y respuesta es similar a la de mensajes de correo electrónico; consisten en un grupo de líneas que contienen cabeceras de los mensajes, seguido por una línea en blanco, seguido por un cuerpo de mensaje. El servidor al recibir esta solicitud intenta generar una respuesta HTTP que debe ser interpretada por el navegador como HTML y despliega su contenido.

HTTPS

Es la versión segura del protocolo HTTP, utilizado para la transferencia de la información de forma segura en el internet.

Todos los mensajes HTTPS son los mismos excepto que se transmiten a través de Secure Socket Layer (SSL)¹, los cuales son encriptados antes de la transmisión y descifrados después de haber sido recibidos por el servidor (Leon Shklar, 2003).

En HTTPS, el cliente abre una conexión al puerto 443 (el puerto por defecto para HTTP seguro) en el servidor web. Una vez que la conexión se establece, el cliente y el servidor inicializan la capa SSL, para negociar los parámetros de criptografía y el intercambio de claves. Luego de esto la inicialización de SSL se lleva a cabo, y el cliente puede enviar mensajes de solicitud a la capa de seguridad. Estos mensajes están encriptados antes de ser enviados.

¹¹ Protocolo de cifrado que proporciona comunicaciones seguras en una red



HTML (Hyper Text Markup Language)

Es el lenguaje de marcado de hipertexto que define la sintaxis y conjunto de instrucciones para la creación de páginas web. Estos lenguajes hacen un documento interactivo por medio de enlaces que conectan a documentos o recursos de internet.

HTML proporciona muchas maneras diferentes que permiten definir el aspecto de los documentos, centrándose en la estructura no en la apariencia. De hecho, contiene muchas maneras para la estructuración de su contenido: los encabezados de sección, las listas estructuradas, párrafos, reglas, títulos, imágenes incrustadas y son definidos por los lenguajes estándar sin tener en cuenta cómo estos elementos pueden ser presentados por un navegador. Sin embargo, la apariencia es importante, es por eso que se utiliza CCS2. (Kennedy Bill, 2006)

HTML contiene etiquetas que permiten indicar la semántica del contenido del documento, algo que está incompleto o con frecuencia mal implementado en los procesadores de texto y programas de diseño de páginas.

Lenguaje de programación C#

C# es un lenguaje de programación creado por Microsoft para crear programas para su plataforma .NET, pero estandarizado posteriormente por ECMA y por ISO, y del que existe una implementación alternativa de “código abierto”, el “proyecto Mono”, que está disponible para Windows, Linux, Mac OS X y otros sistemas operativos. (Cabanés, 2012)

Características de C#:

- Sencillez de uso.
- Orientado a objetos.
- Orientado a componentes.
- Recolección de basura.
- Instrucciones seguras.
- Unificación de tipos.
- Extensión de los operadores básicos.
- Extensión de modificadores.

² Hojas de Estilo en Cascada



Bootstrap

Es un framework o conjunto de herramientas de software libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales.
https://librosweb.es/libro/ajax/capitulo_1.html (Libros Web)

JQuery

JQuery es una librería JavaScript open-source, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, eventos, efectos dinámicos y crear aplicaciones interactivas con la técnica AJAX. Además, cuenta con la posibilidad de agregar plugins, facilitando aún más nuestro trabajo.

Características:

- Es flexible y rápido para el desarrollo web.
- Funciona en múltiples navegadores.
- Compatible con CSS3.
- Soporta extensiones.
- Permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX.

AJAX

AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano. (Libros Web)

Almacenamiento y Acceso a Datos

SQLite

SQLite es una biblioteca escrita en lenguaje C que implementa un Sistema de gestión de bases de datos transaccionales SQL auto-contenido, sin servidor y sin configuración. SQLite emplea registros de tamaño variable de forma tal que se utiliza el espacio en disco que es realmente necesario en cada momento. (EcuRed, s.f.)

SQLite es un sistema completo de bases de datos que soporta múltiples tablas, índices, triggers y vistas. No necesita un proceso separado funcionando como servidor ya que lee



y escribe directamente sobre archivos que se encuentran en el disco duro. El formato de la base de datos es multiplataforma e indistintamente se puede utilizar el mismo archivo en sistemas de 32 y 64 bits.

SQL SERVER

Microsoft SQL Server es un sistema de administración y análisis de bases de datos relacionales de Microsoft para soluciones de comercio electrónico, línea de negocio y almacenamiento de datos. (Microsoft, s.f.)

Características:

- Soporte de transacciones.
- Soporta procedimientos almacenados.
- Incluye también un entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Además, permite administrar información de otros servidores de datos.

ADO.NET Entity Framework

El ADO.NET Entity Framework es un conjunto de APIs de acceso a datos para el Microsoft .NET Framework, apuntando a la versión de ADO.NET que se incluye con el .NET Framework.

Una entidad del Entity Framework es un objeto que tiene una clave representando la clave primaria de una entidad lógica de datastore. Un modelo conceptual Entity Data Model (modelo Entidad-Relación) es mapeado a un modelo de esquema de datastore. Usando el Entity Data Model, el Framework permite que los datos sean tratados como entidades independientemente de sus representaciones del datastore subyacente.

El Entity SQL es un lenguaje similar al SQL para consultar el Entity Data Model (en vez del datastore subyacente). Similarmente, las extensiones del Linq, Linq-to-Entities, proporcionan consultas tipeadas en el Entity Data Model. Las consultas Entity SQL y Linq-to-Entities son convertidas internamente en un Canonical Query Tree que entonces es



convertido en una consulta comprensible al datastore subyacente (ej. en SQL en el caso de una base de datos relacional). Las entidades pueden utilizar sus relaciones, y sus cambios enviados de regreso al datastore.

SQL Reporting Services (SSRS)

SQL Server Reporting Services es una solución que los clientes pueden implementar de forma local para crear, publicar y administrar informes y, luego, entregarlos a los usuarios adecuados de distintas formas: visualizándolos en el explorador web, en el dispositivo móvil o como correo electrónico en su Bandeja de entrada. (Microsoft.com, 2016)

Un servidor de informes es un equipo que tiene instalada una instancia de Reporting Services.

Un servidor de informes almacena internamente elementos como informes paginados y móviles, elementos y recursos relacionados con informes, programaciones y suscripciones. Un servidor de informes se puede configurar como un único servidor independiente o como una granja escalada, o se puede integrar con SharePoint Server. Se interactúa con los elementos del servidor de informes mediante el servicio web Reporting Services, un proveedor WMI, acceso mediante dirección URL o mediante programación a través de scripts. La forma en que interactúa con un servidor de informes depende de la topología de implementación y la configuración.

Ventajas de los informes paginados de Reporting Services.

Se puede usar las soluciones de informes de Reporting Services para:

- Usar un solo conjunto de orígenes de datos que proporcione una única versión de los hechos. Puede basar los informes en esos orígenes de datos para proporcionar una vista de datos unificada que facilite la toma de decisiones comerciales.
- Visualizar los datos de formas diversas e interconectadas a través de las regiones de datos. Puede mostrar los datos organizados en tablas, matrices o tablas de referencias cruzadas; también puede expandir o contraer grupos, gráficos,



medidores, indicadores o KPI, y mapas, e incluso tiene la posibilidad de incluir gráficos en las tablas.

- Ver los informes para su propio uso o publicar informes en un servidor de informes o un sitio de SharePoint para compartirlos con el equipo o la organización.
- Definir un informe una sola vez y presentarlo de diversas maneras. Puede exportar el informe en varios formatos de archivo o entregar el informe a los suscriptores en forma de correo electrónico o de un archivo compartido. Puede crear informes vinculados que apliquen distintos conjuntos de parámetros a la misma definición de informe.
- Usar elementos de informe, orígenes de datos compartidos, consultas compartidas y sub-informes para definir las visualizaciones de datos para su reutilización.
- Administrar los orígenes de datos del informe con independencia de la definición de informe. Por ejemplo, puede cambiar de un origen de datos de prueba a un origen de datos de producción sin cambiar el informe.
- Crear informes con un diseño libre. El diseño del informe no está restringido a bandas de información. Puede organizar la visualización de los datos de la página de forma que facilite su comprensión, mejore su entendimiento y promueva la entrada en acción.
- Habilitar acciones para la obtención de detalles, alternadores para expandir y contraer, botones de ordenación, información sobre herramientas y parámetros de informe que permitan al lector interactuar con el informe. Puede combinar los parámetros de informe con sus propias expresiones para que los lectores del informe puedan controlar el modo en que se filtran, agrupan y ordenan los datos.
- Definir expresiones que le proporcionan la capacidad de personalizar el modo en que se filtran, agrupan y ordenan los datos.



Metodología de Desarrollo

RAD como metodología de desarrollo

El desarrollo rápido de aplicaciones (RAD) es un proceso de desarrollo de software, planteado inicialmente por James Martin en 1980. El método comprende el desarrollo iterativo, la construcción de prototipos y el uso de herramientas CASE.

Fases del RAD

El proceso RAD utiliza componentes de programas ya existentes (cuando es posible) o crea componentes reutilizables (cuando sea necesario). En todos los casos se utilizan herramientas automáticas para facilitar la construcción del software desglosado en fases.

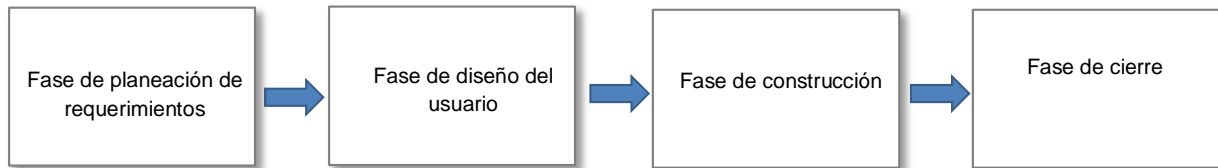


Ilustración 2: Fases de diseño RAD

- **Fase de planificación de Requerimientos:** Martin explica la planeación de requerimientos. Aquí los usuarios de alto nivel deciden qué funciones debe incluir la aplicación.
- **Fase de diseño de usuario:** En esta Martin caracteriza a los usuarios como ocupados en discutir los aspectos no técnicos del diseño del sistema, con la ayuda de los analistas.
- **Fase de construcción:** Acá se realizan muchas actividades diferentes. Tan pronto como las nuevas funciones están disponibles, se muestran a los usuarios para la interacción, comentarios y revisión. Con las herramientas CASE, los analistas pueden hacer cambios continuos en el diseño de las aplicaciones.
- **Fase de cierre:** La aplicación recientemente desarrollada reemplazará a la anterior. Mientras está ejecutándose en paralelo con la aplicación anterior, la nueva se prueba, los usuarios son entrenados y los procedimientos de la organización se cambian antes de que ocurra el cierre.



Prototipos de sistema

Prototipos de sistemas de información es una técnica valiosa para recopilar rápidamente información específica sobre los requerimientos de información de los usuarios. (Kendall, 2011)

Características de los Prototipos:

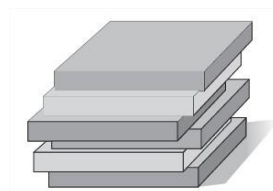
- Se crean con rapidez.
- Evolucionan a través de un proceso iterativo.
- Tienen un costo bajo de desarrollo.

Tipos de prototipo:

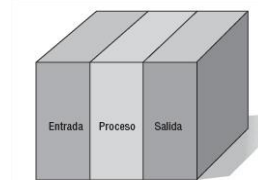
Prototipo primero de una serie: es la creación de un modelo a escala completa de un sistema, a lo que comúnmente se le conoce como piloto. El prototipo es completamente funcional; es una realización de lo que el diseñador espera sea una serie de aeroplanos con características idénticas.

Prototipo de parches: alude a la construcción de un sistema funcional, parchado o construido totalmente con parches, se trata de un modelo funcional, con todas las características necesarias, pero que es ineficiente, debido a que los programas se escribieron con rapidez, con el objetivo de que fuera funcional en vez de eficiente.

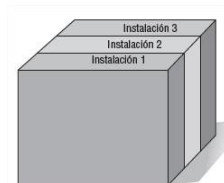
Prototipo no operacional: es el modelo a escala no funcional, empleado para probar ciertos aspectos del diseño.



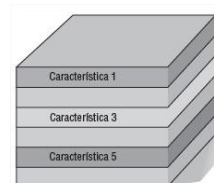
Prototipo de parche



Prototipo no operacional



Prototipo primero de una serie



Prototipo de características selectas

Ilustración 3: Tipos de prototipo



Prototipo de características selectas: es la creación de un modelo operacional que incluya sólo algunas características del sistema final. El sistema se desarrolla en módulos, de manera que, si los usuarios evaluaron positivamente las características presentadas, se pueden incorporar al sistema final sin tener que trabajar mucho para interconectar los módulos.

El tipo de prototipo que se desarrolló en el proyecto es el prototipo primero de serie debido a que el prototipo será completamente funcional como sistema con modelo a escala completa, siendo utilizado como una modalidad alternativa para búsqueda de piezas de refacción.



Análisis y presentación de resultados

Se presenta el análisis y resultado de la propuesta de desarrollo del prototipo de sistema de realidad virtual para la caracterización de piezas automotrices. En este capítulo se dan a conocer las interpretaciones de los datos obtenidos por medio de los métodos y herramientas seleccionados para el desarrollo de los prototipos.

En el análisis, se elabora los requerimientos del prototipo de sistema, el modelo de gestión, modelo de datos y modelo de procesos. En el diseño se presentan todas las interfaces entre el sistema y el usuario, tales como formularios de entrada, de uso del sistema, generación de proforma.

Fase de planificación de Requerimientos

En esta fase de plantean los requerimientos necesarios para el desarrollo del prototipo de sistema de realidad virtual.

Requerimientos funcionales

Requerimiento	Prioridad
Registro de usuarios en el prototipo de sistema de realidad virtual.	Esencial
Registrar y editar catálogos (vehículos, marcas, etc.)	Esencial
Modelos de piezas en 3D	Esencial
La búsqueda de piezas se debe poder realizar de 3 maneras: Por Vehículo Por Código Por Nombre de pieza	Esencial
Realizar transformaciones básicas sobre una pieza (mover, rotar, escalar)	Esencial
Mostrar las especificaciones de las piezas vehiculares.	Esencial
Indicar el estado de la pieza en la que se encuentra en inventario, distinguiendo su disponibilidad.	Esencial
Generar proforma	Esencial
Las proformas se deben exportar en distintos formatos (pdf, word, Excel, etc.).	Esencial
Registro entrada y salida de productos	Esencial
Generar reportes de entradas y salidas	Opcional
Migración de productos, clientes y proveedores	Esencial
Registro y actualización de datos de empleados	Esencial
Registro y actualización de datos de productos	Esencial



Requerimiento	Prioridad
Registro y actualización de datos de cliente	Esencial
Registro y actualización de datos de vehículos	Esencial
Registro y actualización de datos de bodegas	Esencial
Registro y actualización de datos de precios	Esencial
Mostrar listado de cotización por usuario	Deseable

Tabla 1: Tabla de Requerimientos funcionales

Requerimientos no funcionales

1. De Apariencia e Interfaz

- Los colores de la empresa deben estar presente en el fondo de la aplicación
- Logo de la empresa al iniciar la aplicación.
- El Icono de la Aplicación deberá representar claramente al software.

2. De Usabilidad

- Manual de usuario.
- Acceso a opciones de Ayuda
- Legibilidad de las tipografías.
- Verificación de ortografía en base al idioma nativo (Español Latinoamericano)

3. De Rendimiento

- El software deberá poder conectarse a la Base de datos cada vez que sea necesario, (Alta disponibilidad)
- Deberá realizar los cálculos en tiempo aceptables con un máximo de 3000 milisegundo (Evitar sobrecarga de datos).
- El consumo del software en memoria física no debe exceder a 1gb RAM para no causar una sobrecarga del equipo.
- El consumo de memoria de video del equipo no debe exceder a la integrada para no afectar el rendimiento de las aplicaciones del equipo de cómputo.
- Un usuario experimentado debe ser capaz de utilizar todas las funciones del sistema tras un entrenamiento de 2 horas, tras el cual no cometerá más de 3 errores diarios.
- Cuando haya hasta 100 usuarios accediendo simultáneamente al sistema, su tiempo de respuesta no será en ningún momento superior a 3000 milisegundos.



- Ante un fallo en el software del sistema, no se tardará más de 5 minutos en restaurar los datos del sistema (en un estado válido) y volver a poner en marcha el sistema.

4. De Confiabilidad

- El software garantiza la integridad y confiabilidad de la información expuesta y compartida al público.

5. De Hardware

- Para hacer uso de SRVSA se debe tener en cuenta los siguientes requerimientos mínimos:
- Procesador mínimo 2.0ghz Core I3
- 2Gb RAM.

6. De Software

- Se debe contar con sistema operativo Windows con el JRE actualizado, que contiene la JVM.

7. De Seguridad

- Acceso al sistema mediante credenciales únicas otorgadas al usuario.
- El usuario Administrador será el único que tendrá los privilegios de modificación de permisos de usuarios.
- Los permisos a los usuarios estarán segregados en roles, roles que serán definidos por el desarrollador del sistema con la autorización del gerente general de la empresa “San Agustín”.
- La conexión a la Base de Datos se hará de forma local (No permitir conexiones desde internet)

Estructura del Sistema (paquetes)

El sistema de la empresa “San Agustín”, posee los siguientes paquetes.

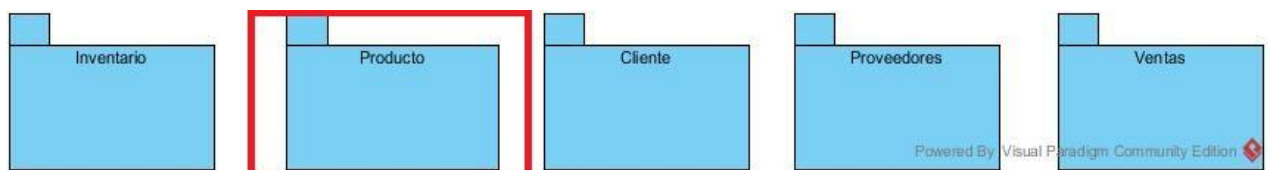


Ilustración 4: Diagrama de paquetes del Sistema de Ventas Repuestos San Agustín.



El prototipo se integrará con el sistema de Ventas Repuestos San Agustín tomando la información de los productos para ser administrados por los paquetes del prototipo de sistema.

El prototipo está dividido en cuatro paquetes que se construirán secuencialmente.

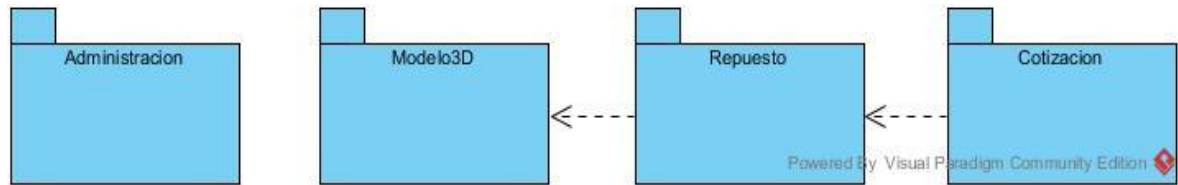


Ilustración 5: Diagrama de paquetes

- Paquete administración: gestiona los permisos y roles de los usuarios del sistema, así como se encarga del mantenimiento de los catálogos propios del sistema.
- Paquete Modelo3D: contiene las piezas vehiculares del sistema SRVSA.
- Paquete Repuesto: construye un modelo lógico y funcional capaz de cumplir con las funciones delegadas a la búsqueda y visualización a detalle de piezas seleccionadas.
- Paquete Cotización: genera el documento que representa la pre-venta de las piezas que se vayan posteriormente a adquirir.

Calendarización (asignación de recursos)

La etapa de calendarización está dividida en la siguiente línea de tiempo. Detallada a continuación:



Ilustración 6: Interacción entre las Fases del RAD



Como el prototipo fue construido para usuarios con diferentes niveles de desarrollo en el área de ventas de auto repuestos se avanzará en su construcción iterando las fases de diseño de usuario y construcción las veces que sean necesarias para lograr que la fase de construcción recopile todos los requisitos de los usuarios.

Es posible que los requerimientos Fase de planificación de Requerimientos pag21 de este documento queden sujetos a cambios a medida que se identifiquen nuevos requerimientos en las iteraciones de las fases del RAD fundamentados en el prototipaje.

Fase de diseño de usuario

En esta fase se representan el modelo de gestión, que representa las funciones principales del negocio y el modelado de datos mediante los diagramas entidad relación constituido en paquetes.

Modelado de gestión

En este modelo se presentan las áreas administrativas que se encuentran en la empresa “San Agustín”.



Ilustración 7: Modelo de gestión



Modelado de datos

El modelo de datos del prototipo de sistema está dividido en paquetes de la siguiente manera:

Paquete de Administración

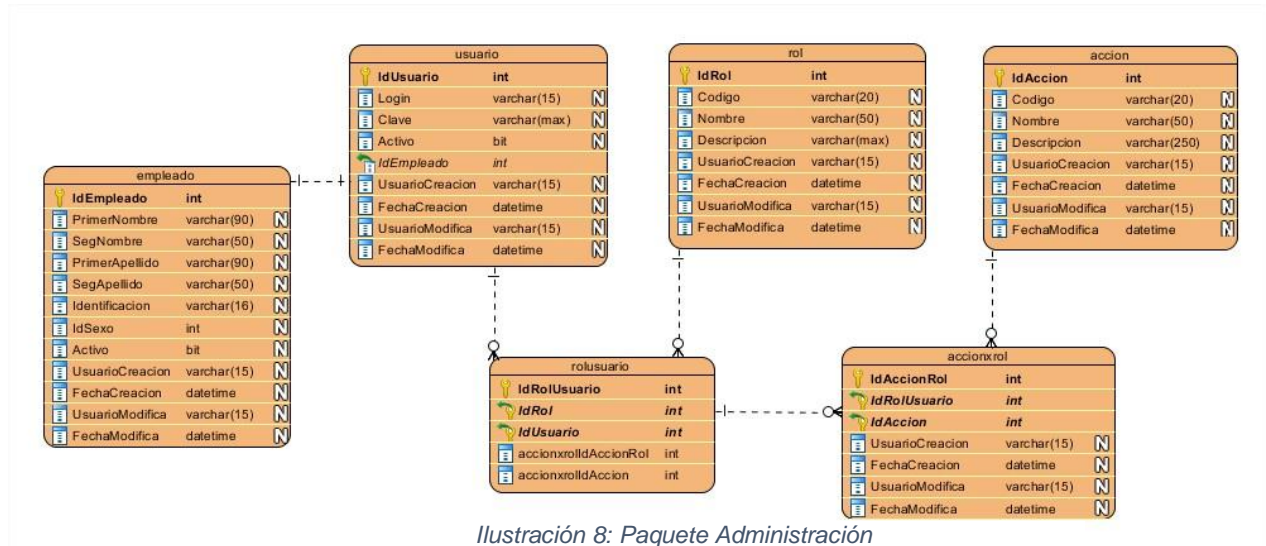


Ilustración 8: Paquete Administración

Paquete M

Este paquete contiene las piezas vehiculares del sistema SRVSA. Dichas piezas serán construidas a través de la técnica de modelaje NURBS ver Técnicas de modelado 3D pag7.

Proceso de modelaje de piezas automotrices

Para iniciar este proceso es necesario contar con alguna imagen fotográfica desde diferentes ángulos, o diagramas de diseño del objeto que se desea modelar en 3D (en nuestro caso de las piezas automotrices); estas imágenes son insertada en el software modelador como fondo de los viewports que se muestran en pantalla donde servirán de guía para el dibujo de contorno usando los curvas NURBS.

Los fondos de los viewports deberán de configurarse de tal manera que estén sincronizados entre sí, y no sufran alteraciones al momento de panear la imagen o realizar acercamiento ya que estos fondos serán los templates que nos permitirán formar la imagen tridimensional. Una vez configurados los fondos se procede a utilizar herramientas CV NURBS, que nos permiten realizar curvas definidas con vértices de control para contornear los bordes del objeto.



Una vez definidos los bordes con CV NURBS se procede a combinar los puntos sueltos de cada vértice usando la opción Fuse que se encuentra en el modificador de la Curva NURBS de este modo tendremos todas las líneas y contornos definidos. Para finalizar el proceso de diseño se necesita crear la superficie que rellena las aristas, para esto en las opciones del modificador de la Curva NURBS buscamos la opción de creación de superficies y seleccionamos la que sea más útil, en este caso particular utilizamos la opción N-BLEND que nos crea la superficie a partir de un enmallado geométrico que parte de los vértices que comprenden la superficie.

El proceso de texturizado es importante para darle el tono realista al diseño del objeto 3D, depende de este proceso que el objeto que se modele en 3D tome un aspecto más realista o un aspecto caricaturesco.

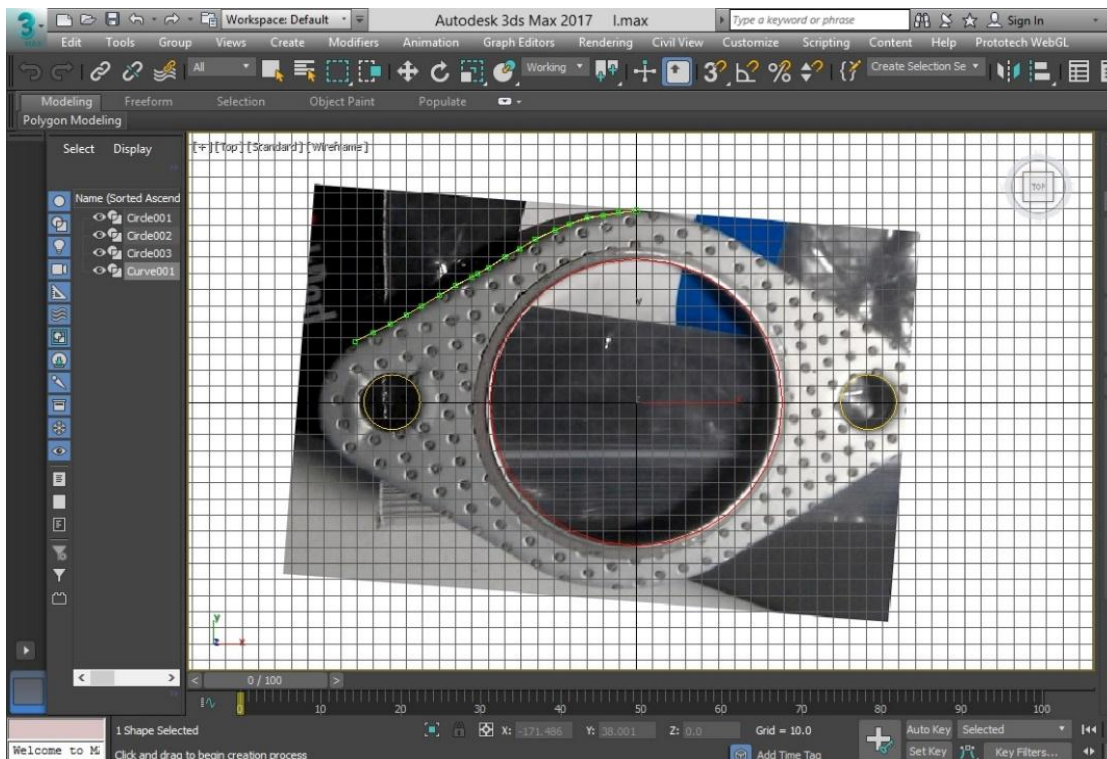


Ilustración 9: Imagen incorporada en software Modelador

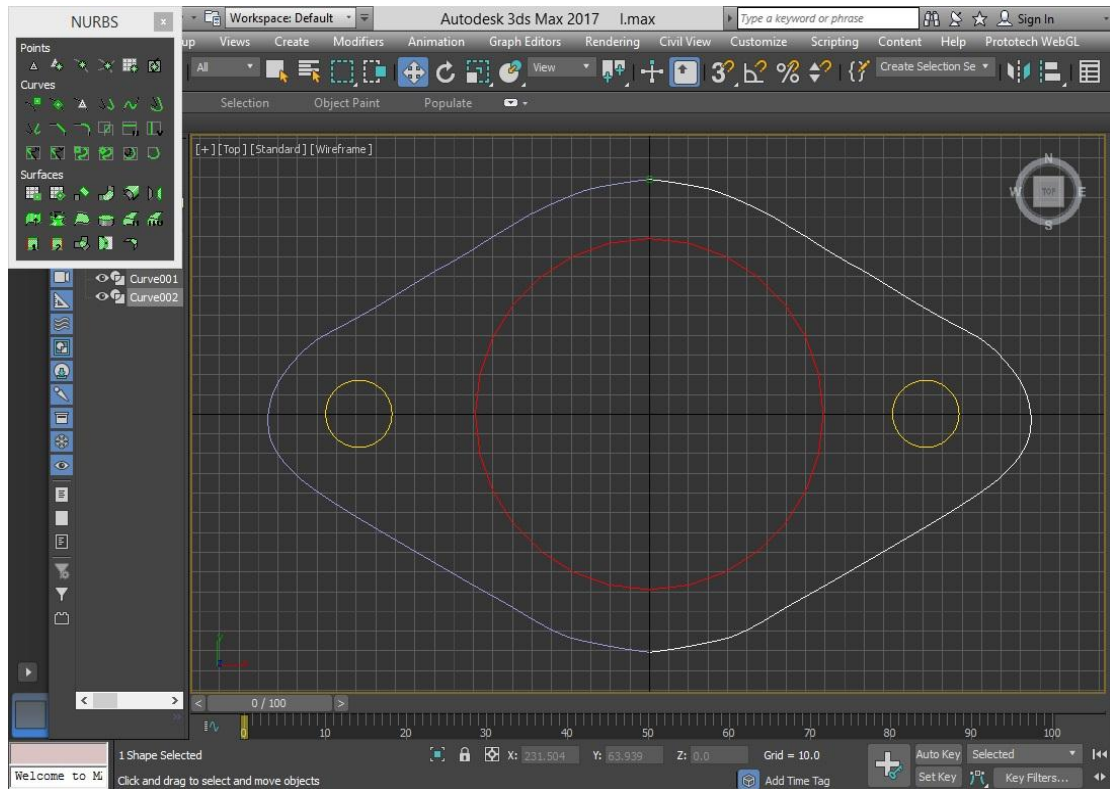


Ilustración 10: Imagen diseñada en 3D con técnica NURBS

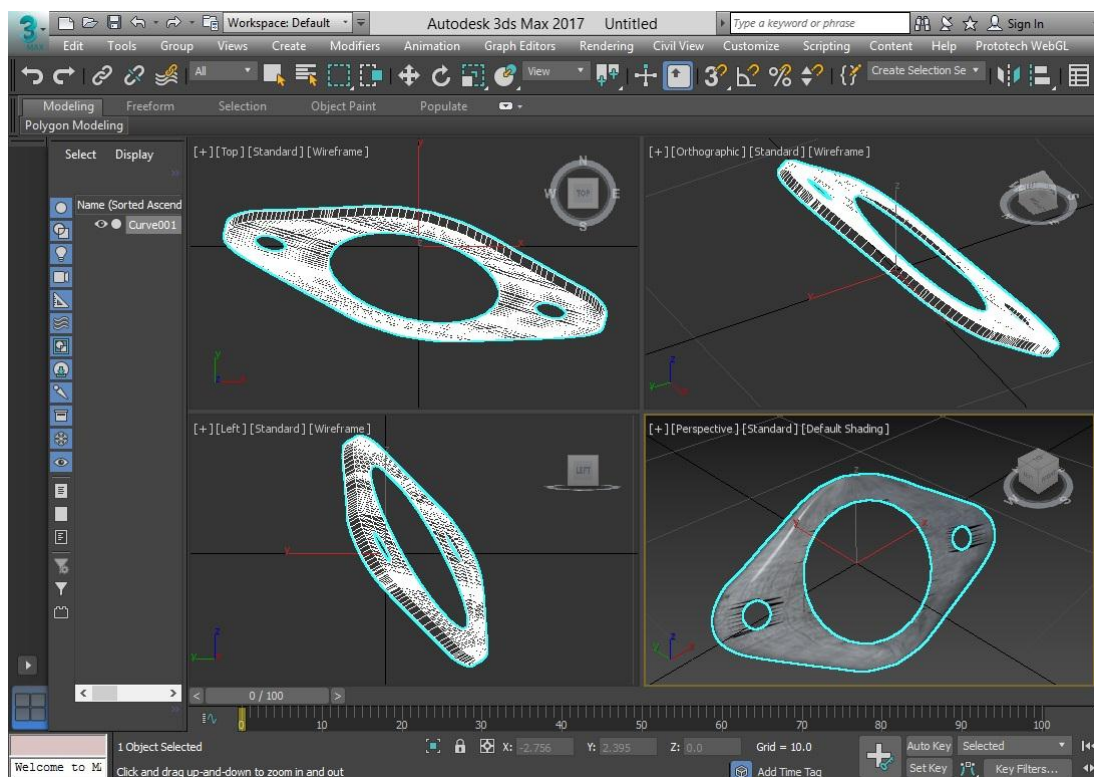


Ilustración 11: Diseño de pieza 3D



Paquete de Repuesto

El paquete de Repuesto toma la información de los productos para mostrar el detalle de las piezas permitiendo posteriormente generar una proforma.

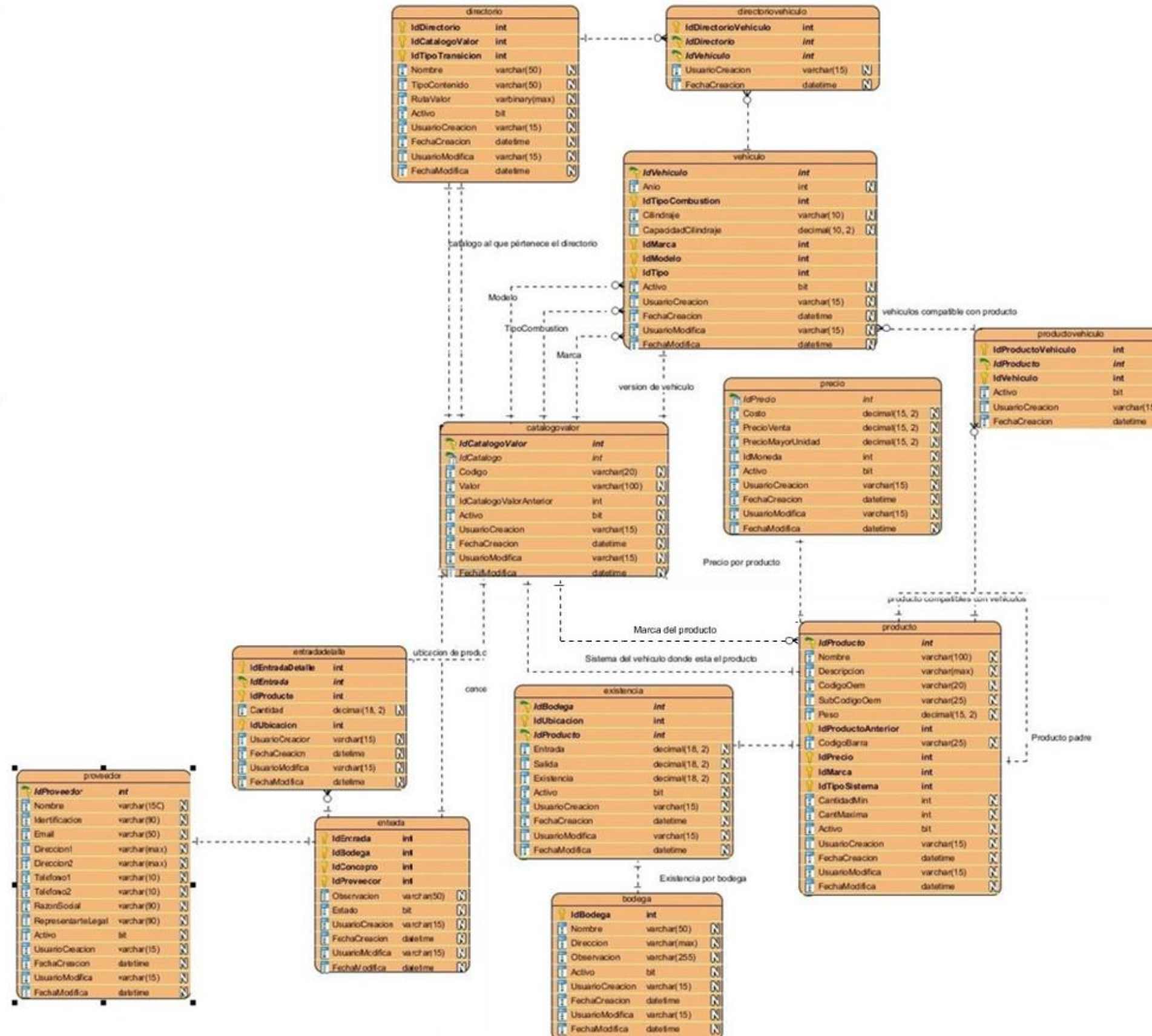


Ilustración 12: Paquete Repuesto

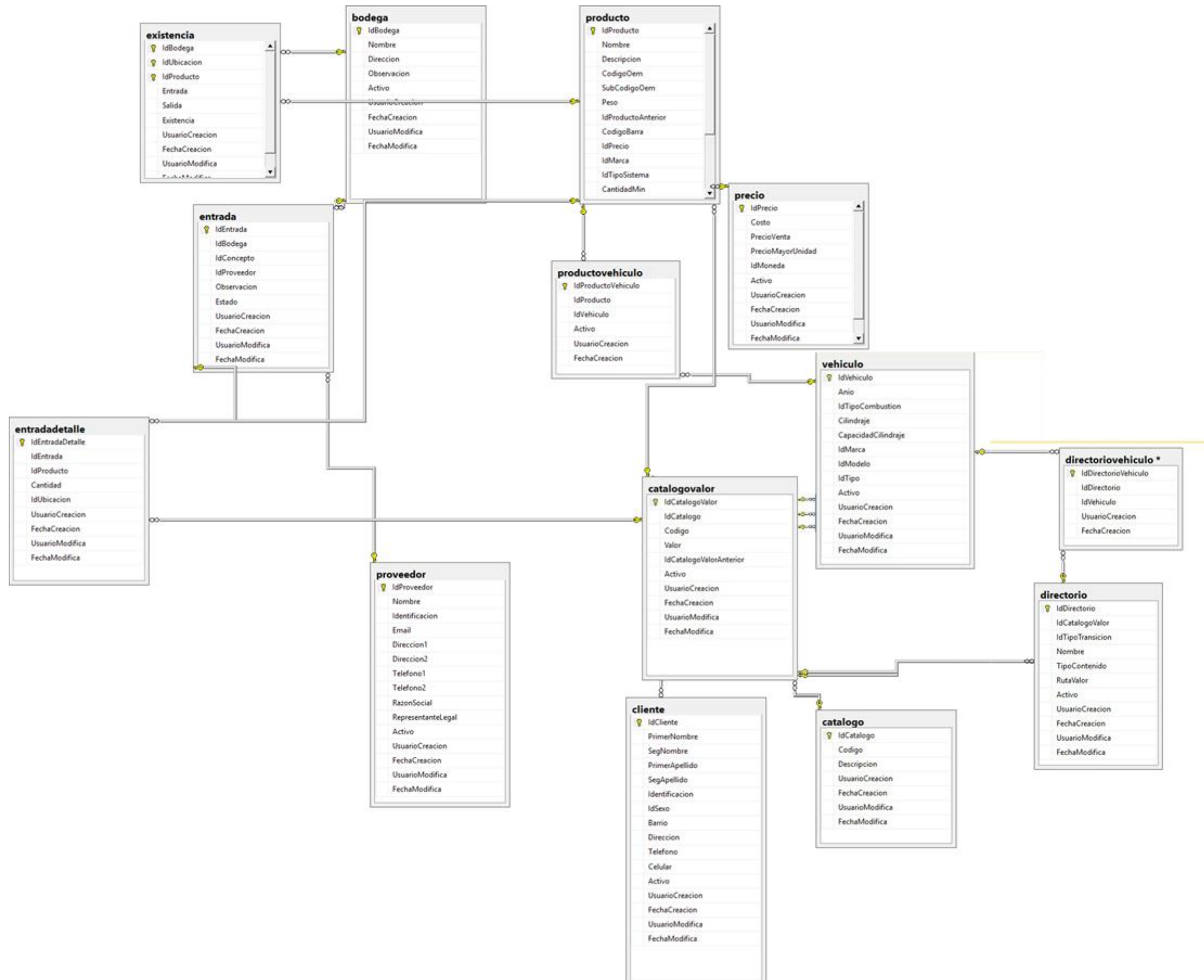


Paquete de Cotización

El paquete de Cotización crea una cotización por cliente tomando una lista de productos con las cantidades a cotizar.



Ilustración 13: Paquete Cotización



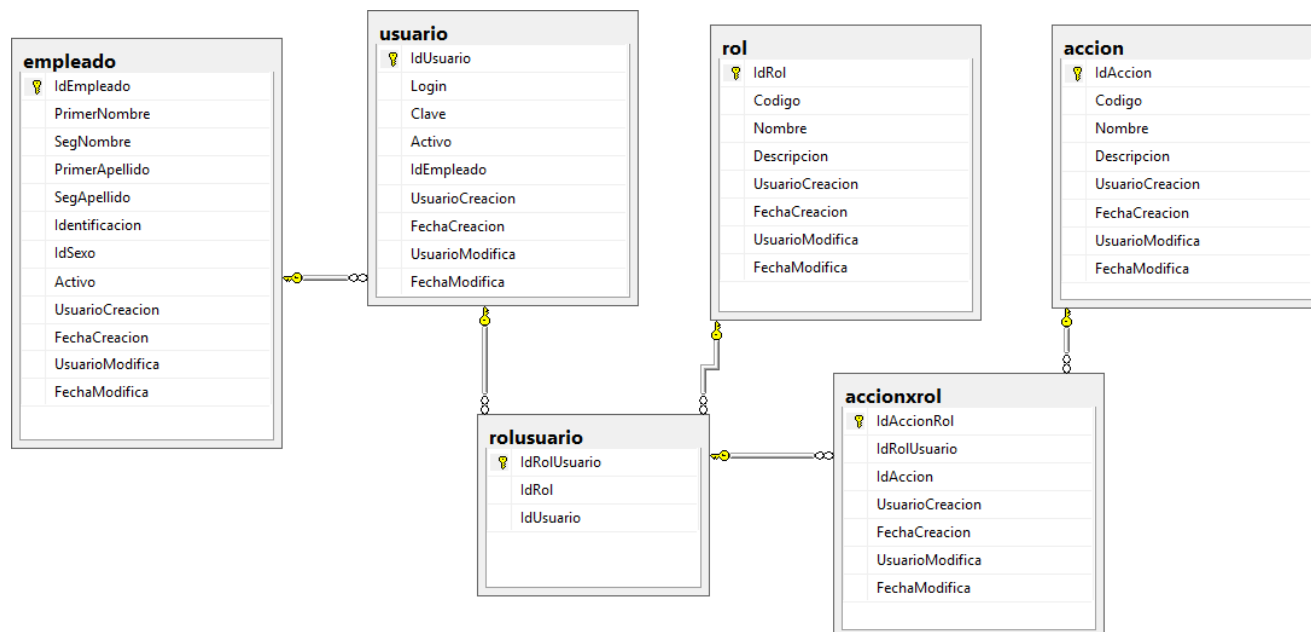


Ilustración 15: Diagrama de Entidad-Relación Seguridad SRVSA

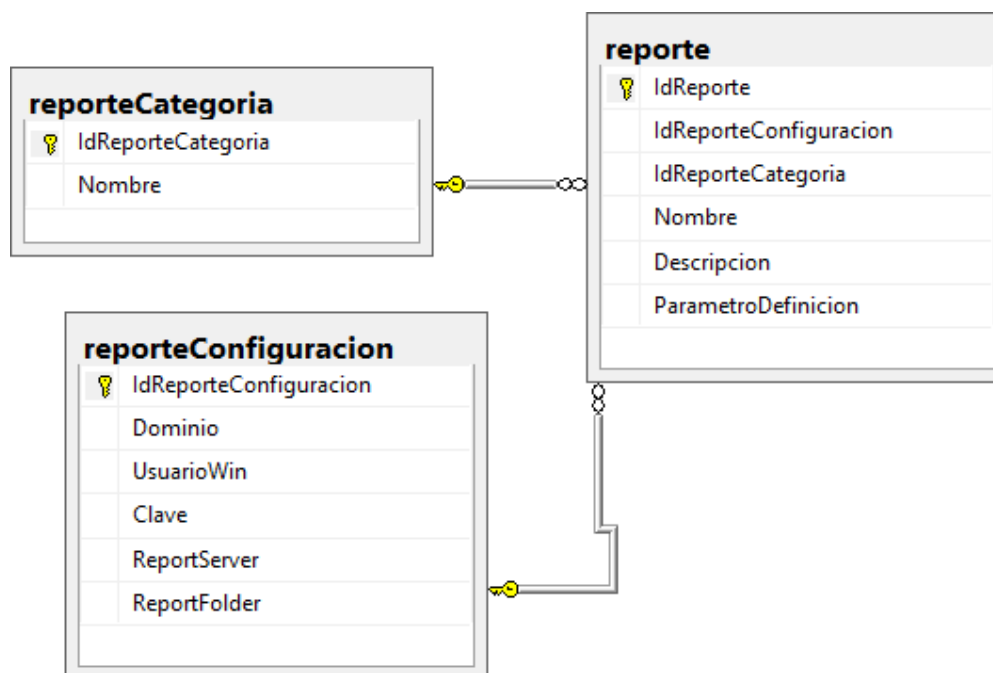


Ilustración 16: Diagrama de Entidad-Relación Configuración Reporte SRVSA

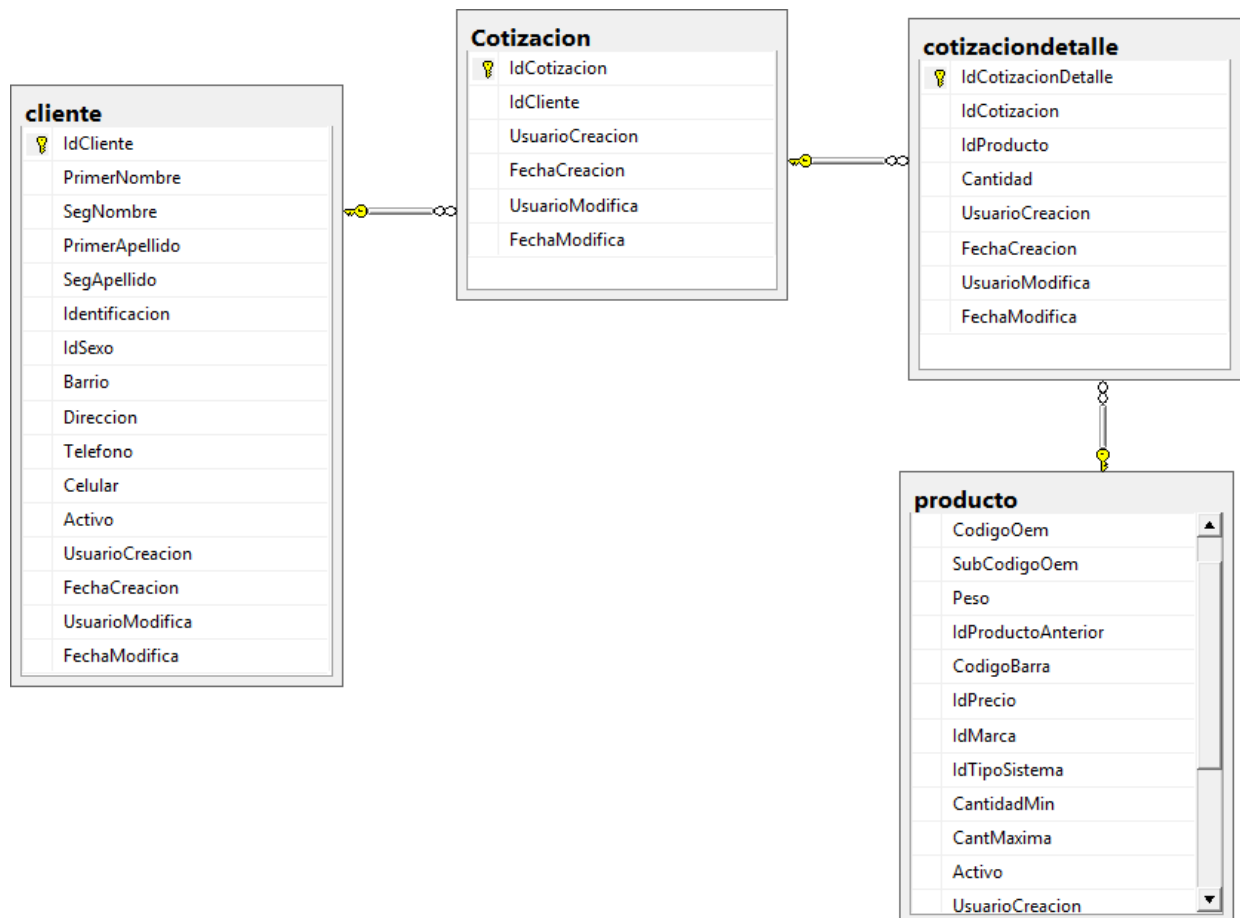


Ilustración 17: Diagrama de Entidad-Relación Cotización SRVSA



Diccionario de datos

Tabla usuario		Posee los usuarios registrados en el sistema		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdUsuario	PK	int		Identificador de usuario
Login		varchar(15)	15	nombre de usuario del sistema
Clave		varchar(MAX)		Contraseña de usuario
Activo		bit		Indica si el usuario se encuentra activo en el sistema
UsuarioCreacion		varchar(15)	15	Campo auditoria para el usuario que crea el registro
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro
UsuarioModifica		varchar (15)	15	Campo auditoria para el usuario que modifica el registro
FechaModifica		datetime		Campo auditoria para la fecha en el que el usuario modifico el registro

Tabla 2: Tabla de base de datos usuario

Tabla rol		Posee los las funciones asignadas para cada usuario		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdRol	PK	int		Identificador de rol
Codigo		varchar(20)	20	Código único asignado al Rol
Nombre		varchar(50)	50	Nombre asignado al Rol
Descripcion		varchar(MAX)		Descripción de la función que desempeñara dicho rol
UsuarioCreacion		varchar(15)	15	Campo auditoria para el usuario que crea el registro
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro
UsuarioModifica		varchar(15)	15	Campo auditoria para el usuario que modifica el registro
FechaModifica		datetime		Campo auditoria para la fecha en el que el usuario modifico el registro

Tabla 3: Tabla de base de datos rol



Tabla rolusuario		Posee los roles asignados para cada usuario		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdRolUsuario	PK	int		Identificador de rol usuario
IdRol	FK	int		Identificador de rol asignado
IdUsuario	FK	int		Identificador de usuario que posee el rol u roles
UsuarioCreacion		varchar(15)	15	Campo auditoria para el usuario que crea el registro
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro

Tabla 4: Tabla de base de datos rolusuario

Tabla accion		Posee las operaciones que podrá realizar el usuario		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdAccion	PK	int		Identificador de accion
Codigo		varchar(20)	20	Código único asignado a la acción
Nombre		varchar(50)	50	Nombre asignado a la accion
Descripcion		varchar(MAX)		Descripcion de la operación
UsuarioCreacion		varchar(15)	15	Campo auditoria para el usuario que crea el registro
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro
UsuarioModifica		varchar(15)	15	Campo auditoria para el usuario que modifica el registro
FechaModifica		datetime		Campo auditoria para la fecha en el que el usuario modifico el registro

Tabla 5: Tabla de base de datos accion

Tabla accioxrol		Posee la lista de acciones asignadas a un rol		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdAccionRol	PK	int		Identificador de accion x rol
IdRolUsuario	FK	int		Identificador de rol por usuario
IdAccion	FK	int		Identificador de la operación que podrá realizar un rol asignado a un usuario
UsuarioCreacion		varchar(15)	15	Campo auditoria para el usuario que crea el registro
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro
UsuarioModifica		varchar(15)	15	Campo auditoria para el usuario que modifica el registro



FechaModifica		datetime		Campo auditoria para la fecha en el que el usuario modifico el registro
---------------	--	----------	--	-------------------------------------------------------------------------

Tabla 6: Tabla de base de datos accionxrol

Posee la lista genérica con el objetivo de minimizar la cantidad de objetos del sistema				
Tabla catalogo				
Campo	Llave	Tipo de datos	Longitud	Descripción
IdCatalogo	PK	int		Identificador de catalogo
Codigo		varchar(25)	25	Código único asignado al catálogo creado
Descripcion		varchar(MAX)		Descripción del catálogo creado
UsuarioCreacion		varchar(15)	15	Campo auditoria para el usuario que crea el registro
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro
UsuarioModifica		varchar(15)	15	Campo auditoria para el usuario que modifica el registro
FechaModifica		datetime		Campo auditoria para la fecha en el que el usuario modifico el registro

Tabla 7: Tabla de base de datos catalogo

Posee los objetos de las listas genérica del sistema				
Tabla catalogovalor				
Campo	Llave	Tipo de datos	Longitud	Descripción
IdCatalogoValor	PK	int		Identificador del objeto de la lista genérica
IdCatalogo	FK	int		Identificador de la lista genérica
Codigo		varchar(20)	20	Código único asignado al objeto de la lista genérica
Valor		varchar (100)	100	Descripción corta del objeto a crear
IdCatalogoValorAnterior	FK	int		Identificador Padre cuando la lista de objetos pertenece a una lista existente
UsuarioCreacion		varchar (15)	15	Campo auditoria para el usuario que crea el registro
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro
UsuarioModifica		varchar(15)	15	Campo auditoria para el usuario que modifica el registro
FechaModifica		datetime		Campo auditoria para la fecha en el que el usuario modifico el registro

Tabla 8: Tabla de base de datos catalogovalor



Tabla directorio		Contiene las imágenes usadas para las animaciones del sistema		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdDirectorio	PK	int		Identificador del contenedor de archivos
IdCatalogoValor	FK	int		Identificador del nombre del objeto al que hace referencia la imagen
IdTipoTransicion	FK	int		Identificador del objeto al que hace referencia el tipo de animación de la imagen
Nombre		varchar(50)	50	nombre de la imagen
TipoContenido		varchar(50)	50	tipo de archivo de la imagen (jpg, bmp, png, etc.)
RutaValor		varbinary(MAX)		contiene el archivo en formato byte
Activo				Indica si el contenedor de archivo se encuentra activo en el sistema
UsuarioCreacion		varchar(15)	15	Campo auditoria para el usuario que crea el registro
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro
UsuarioModifica		varchar(15)	15	Campo auditoria para el usuario que modifica el registro
FechaModifica		datetime		Campo auditoria para la fecha en el que el usuario modifico el registro

Tabla 9: Tabla de base de datos directorio

Tabla vehiculo		Posee la lista de vehículos del sistema		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdVehiculo	PK	int		Identificador del vehículo
Anio		int		Año de fabricación del vehículo
IdTipoCombustion	FK	int		Identificador de catálogo que indica el tipo de combustión del vehículo
Cilindraje		varchar(10)	10	Indica la potencia del motor expresado en centímetros cúbicos.
CapacidadCilindraje		decimal(10, 2)		Identificador Padre cuando la lista de objetos pertenece a una lista existente
IdMarca	FK	int		Identificador de catálogo de marca de vehículo
IdModelo	FK	int		Identificador de catálogo de los modelo de vehículo



Tabla vehiculo		Posee la lista de vehículos del sistema		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdTipo	FK	int		Identificador de catálogo de la versión del vehículo según marca y modelo
Activo		bit		Indica si el vehículo se encuentra activo en el sistema
UsuarioCreacion		varchar(15)	15	Campo auditoria para el usuario que crea el registro
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro
UsuarioModifica		varchar(15)	15	Campo auditoria para el usuario que modifica el registro
FechaModifica		datetime		Campo auditoria para la fecha en el que el usuario modifico el registro

Tabla 10: Tabla de base de datos vehiculo

Tabla producto		Posee la lista de productos comercializados por la empresa		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdProducto	PK	int		Identificador del producto
Nombre		varchar(100)		
Descripcion		varchar(MAX)		Detalle del producto
CodigoOem		varchar(20)	20	Código identificador del producto proporcionado por el fabricante
SubCodigoOem		varchar(25)	25	Número de serie de fabricante aftermarket de tipos de piezas
Peso		decimal(15, 2)		Peso del producto
IdProductoAnterior	FK	int		Identificador del producto padre para los productos que son aftermarket
CodigoBarra		varchar(25)	25	Identificador del producto proporcionado por la empresa
IdPrecio	FK	int		Identificador de precio de catálogo de precio para el comercio del producto
IdMarca	FK	int		Identificador de marca del producto obtenido de catálogo de marcas
IdTipoSistema	FK	int		Identificador del sistema al que pertenece el producto



Tabla producto Posee la lista de productos comercializados por la empresa				
Campo	Llave	Tipo de datos	Longitud	Descripción
CantidadMin		int		Cantidad mínima permitida del producto
CantMaxima		int		Cantidad máxima permitida del producto
Activo		bit		Indica si el producto obtenido desde el sistema se encuentra activo
UsuarioCreacion		varchar(15)	15	Campo auditoria para el usuario que crea el registro
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro
UsuarioModifica		varchar(15)	15	Campo auditoria para el usuario que modifica el registro
FechaModifica		datetime		Campo auditoria para la fecha en el que el usuario modifico el registro

Tabla 11: Tabla de base de datos producto

Tabla productovehiculo Posee la lista de vehículo compatibles con la lista de productos del sistema				
Campo	Llave	Tipo de datos	Longitud	Descripción
IdProductoVehiculo	PK	int		Identificador autogenerado
IdProducto	FK	int		Identificador de producto obtenido del catálogo de productos
IdVehiculo	FK	int		Identificador de vehículo obtenido del catálogo de vehículos
UsuarioCreacion		varchar(15)	15	Campo auditoria para el usuario que crea el registro
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro
UsuarioModifica		varchar(15)	15	Campo auditoria para el usuario que modifica el registro
FechaModifica		datetime		Campo auditoria para la fecha en el que el usuario modifico el registro

Tabla 12: Tabla de base de datos productovehiculo



Tabla precio		Posee la lista de precios de los productos		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdPrecio	PK	int		Identificador del objeto precio
Costo		decimal(15, 2)		precio de compra del producto
PrecioVenta		decimal(15, 2)		precio de oferta del producto
PrecioMayorUnidad		decimal(15, 2)		precio al por mayor de oferta del producto
IdMoneda	FK	int		Identificador de la moneda del precio de venta de producto
Activo		bit		Indica si el precio obtenido desde el sistema se encuentra activo
UsuarioCreacion		varchar(15)	15	Campo auditoria para el usuario que crea el registro
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro
UsuarioModifica		varchar(15)	15	Campo auditoria para el usuario que modifica el registro
FechaModifica		datetime		Campo auditoria para la fecha en el que el usuario modifico el registro

Tabla 13: Tabla de base de datos precio

Tabla bodega		Posee la lista de bodegas donde se almacena el producto		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdBodega	PK	int		Identificador de la bodega
Nombre		varchar(50)	50	Nombre de la bodega
Direccion		varchar(MAX)		Ubicación exacta de la bodega
Observacion		varchar(255)	255	Información adicional de la bodega
Activo				Indica si la bodega obtenida desde el sistema se encuentra activo
UsuarioCreacion		varchar(15)	15	Campo auditoria para el usuario que crea el registro
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro
UsuarioModifica		varchar(15)	15	Campo auditoria para el usuario que modifica el registro
FechaModifica		datetime		Campo auditoria para la fecha en el que el usuario modifico el registro

Tabla 14: Tabla de base de datos bodega



Tabla existencia		Posee el detalle de existencias en bodega de los productos		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdBodega	PK	int		Identificador de la bodega
IdUbicacion	FK	int		identificador de la ubicación obtenida del catálogo de ubicaciones
IdProducto	FK	int		Identificador del producto
Entrada		decimal(18, 2)		Cantidad surtida del producto obtenido de la solicitud de entrada
Salida		decimal(18, 2)		Cantidad retirada del producto
Existencia		decimal(18, 2)		Cantidad disponible para ofertar
UsuarioCreacion		varchar(15)	15	Campo auditoria para el usuario que crea el registro
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro
UsuarioModifica		varchar(15)	15	Campo auditoria para el usuario que modifica el registro
FechaModifica		datetime		Campo auditoria para la fecha en el que el usuario modifico el registro

Tabla 15: Tabla de base de datos existencia

Tabla proveedor		Posee la lista de abastecedores de los productos		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdProveedor	PK	int		Identificador de proveedor
Nombre		varchar(150)	150	Nombre del proveedor
Identificacion		varchar(90)	90	Número de identificación del proveedor
Email		varchar(50)	50	correo electrónico del proveedor
Direccion1		varchar(MAX)		Ubicación exacta del proveedor
Direccion2		varchar(MAX)		Ubicación exacta del proveedor
Telefono1		nchar(10)	10	numero de contacto del proveedor
Telefono2		nchar(10)	10	numero de contacto del proveedor
RazonSocial		varchar(90)	90	Nombre por el que es conocido colectivamente el proveedor
RepresentanteLegal		varchar(90)	90	Nombre del representante legal del proveedor
Activo		bit		Indica si la bodega obtenida desde el sistema se encuentra activo



Tabla proveedor		Posee la lista de abastecedores de los productos		
Campo	Llave	Tipo de datos	Longitud	Descripción
UsuarioCreacion		varchar(15)	15	Campo auditoria para el usuario que crea el registro
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro
UsuarioModifica		varchar(15)	15	Campo auditoria para el usuario que modifica el registro
FechaModifica		datetime		Campo auditoria para la fecha en el que el usuario modifico el registro

Tabla 16: Tabla de base de datos proveedor

Tabla entrada		Posee el detalle de la solicitud de abastecimiento del producto		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdEntrada	PK	int		Identificador de la entrada
IdBodega	FK	int		identificador del catálogo de bodega
IdConcepto	FK	int		Identificador del concepto por el cual se hará la entrada
IdProveedor	FK	int		Identificador del proveedor del cual se hará la entrada de ser entrada por abastecimiento
Observacion		varchar(250)	250	Información adicional de la solicitud de entrada
Estado		decimal(18, 2)	bit	Estado en el que se encuentra la solicitud el cual solo se activara hasta que ocurra el abastecimiento
UsuarioCreacion		varchar(15)	15	Campo auditoria para el usuario que crea el registro
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro
UsuarioModifica		varchar(15)	15	Campo auditoria para el usuario que modifica el registro
FechaModifica		datetime		Campo auditoria para la fecha en el que el usuario modifico el registro

Tabla 17: Tabla de base de datos entrada

Tabla entradadetalle		Posee la lista de productos solicitadas para abastecer		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdEntradaDetalle	PK	int		Identificador autogenerado
IdEntrada	FK	int		Identificador de la solicitud de entrada



Tabla entradadetalle		Posee la lista de productos solicitadas para abastecer		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdProducto	FK	int		Identificador del catálogo de producto
Cantidad		decimal(18, 2)		Cantidad exacta que se solicita se abastezca
IdUbicacion		int		Identificador de catálogo de ubicación donde se encuentra el producto
UsuarioCreacion		varchar(15)	15	Campo auditoria para el usuario que crea el registro
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro
UsuarioModifica		varchar(15)	15	Campo auditoria para el usuario que modifica el registro
FechaModifica		datetime		Campo auditoria para la fecha en el que el usuario modifico el registro

Tabla 18: Tabla de base de datos entradadetalle

Tabla cliente		Posee la lista de clientes de la empresa		
Campo	Llave	Tipo de datos	Longitud	Descripcion
IdCliente	PK	int		Identificador de cliente
PrimerNombre		varchar(90)	90	Primer nombre del cliente
SegNombre		varchar(50)	50	Segundo nombre de cliente
PrimerApellido		varchar(90)	90	Primer apellido del cliente
SegApellido		varchar(50)	50	Segundo apellido de cliente
Identificacion		varchar(90)	90	Número de identificación del cliente
IdSexo		int		Identificador de catálogo sexo
Barrio		varchar(100)	100	Barrio donde habita el cliente
Direccion		varchar(MAX)		Ubicación exacta del cliente
Telefono		nchar(10)	10	numero de contacto del cliente
Celular		nchar(10)	10	numero de contacto del cliente
Activo		bit		Indica si la bodega obtenida desde el sistema se encuentra activo
UsuarioCreacion		varchar(15)	15	Campo auditoria para el usuario que crea el registro



Tabla cliente		Posee la lista de clientes de la empresa		
Campo	Llave	Tipo de datos	Longitud	Descripción
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro
UsuarioModifica		varchar(15)	15	Campo auditoria para el usuario que modifica el registro
FechaModifica		datetime		Campo auditoria para la fecha en el que el usuario modifico el registro

Tabla 19: Tabla de base de datos cliente

Tabla empleado		Posee la lista de empleados que laboran en la empresa		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdEmpleado	PK	int		Identificador de empleado
PrimerNombre		varchar(90)	90	Primer nombre del empleado
SegNombre		varchar(50)	50	Segundo nombre de empleado
PrimerApellido		varchar(90)	90	Primer apellido del empleado
SegApellido		varchar(50)	50	Segundo apellido de empleado
Identificacion		varchar(90)	90	Número de identificación del empleado
IdSexo		int		Identificador de catálogo sexo
Activo		bit		Indica si la bodega obtenida desde el sistema se encuentra activo
UsuarioCreacion		varchar(15)	15	Campo auditoria para el usuario que crea el registro
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro
UsuarioModifica		varchar(15)	15	Campo auditoria para el usuario que modifica el registro
FechaModifica		datetime		Campo auditoria para la fecha en el que el usuario modifico el registro

Tabla 20: Tabla de base de datos empleado

Tabla reporte		Posee la lista de reportes del sistema		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdReporte	PK	int		Identificador de reporte
IdReporteConfiguracion	FK	int		Identificador de la configuración del servidor de reporte



Tabla reporte		Posee la lista de reportes del sistema		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdReporteCategoria	FK	int		Identificador del nivel al que pertenece el reporte
Nombre		nvarchar(100)	100	Nombre del reporte
Descripcion		nvarchar(500)	500	Detalle de la función que realiza el reporte
ParametroDefinicion		nvarchar(200)	200	lista de parámetros definidos para el reporte

Tabla 21: Tabla de base de datos reporte

Tabla reporteConfiguracion		Posee la configuración del servidor de reporte		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdReporteConfiguracion	PK	int		Identificador de la configuración del servidor de reporte
Dominio		varchar(50)	50	nombre de dominio donde se encuentra configurado el servidor de reporte
UsuarioWin		varchar(50)	50	Nombre de usuario que tiene permiso de administrador para acceder al servidor de reporte
Clave		varchar(50)	50	Contraseña del usuario que tiene permiso al servidor de reporte
ReportServer		varchar(80)	80	Dirección URL del servidor de reportes
ReportFolder		varchar(80)	80	Nombre de la carpeta donde se encuentra alojados los reportes

Tabla 22: Tabla de base de datos reporteConfiguracion

Tabla reporteCategoria		Posee la lista de jerarquía en la que se encuentran distribuidas los reportes		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdReporteCategoria	PK	int		Identificador de la categoría del reporte
Nombre		varchar(50)	50	Nombre del nivel de los reporte

Tabla 23: Tabla de base de datos reporteCategoria



Tabla cotizacion		Posee la lista de proforma a crear		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdCotizacion	PK	int		Identificador de la proforma
IdCliente	FK	int		Identificador de cliente que se le genera la proforma
UsuarioCreacion		varchar(15)	15	Campo auditoria para el usuario que crea el registro
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro
UsuarioModifica		varchar(15)	15	Campo auditoria para el usuario que modifica el registro
FechaModifica		datetime		Campo auditoria para la fecha en el que el usuario modifico el registro

Tabla 24: Tabla de base de datos cotización

Tabla cotizaciondetalle		Posee la lista de productos a cotizar		
Campo	Llave	Tipo de datos	Longitud	Descripción
IdCotizacionDetalle	PK	int		Identificador de detalle de la cotización
IdCotizacion	FK	int		Identificador de la proforma
IdProducto	FK	int		Nombre asignado al Rol
Cantidad		int		Cantidad del producto a cotizar
UsuarioCreacion		varchar(15)	15	Campo auditoria para el usuario que crea el registro
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro
UsuarioModifica		varchar(15)	15	Campo auditoria para el usuario que modifica el registro
FechaModifica		datetime		Campo auditoria para la fecha en el que el usuario modifico el registro

Tabla 25: Tabla de base de datos cotizaciondetalle



Tabla directoriovehiculo	Posee la lista de vehículos que son compatibles con la lista de imágenes almacenadas			
Campo	Llave	Tipo de datos	Longitud	Descripción
IdDirectorioVehiculo	PK	int		Identificador de directorio vehículo
IdDirectorio	FK	int		Identificador de la lista de imágenes
IdVehiculo	FK	int		Identificador del vehículo al que pertenece la imagen
UsuarioCreacion		varchar(15)	15	Campo auditoria para el usuario que crea el registro
FechaCreacion		datetime		Campo auditoria para la fecha en el que el usuario crea el registro

Tabla 26: Tabla de base de datos directoriovehiculo

Modelado de proceso

Los siguientes diagramas representan los procesos que pueden efectuar el usuario administrador en el prototipo de sistema.

Ingreso al Sistema.

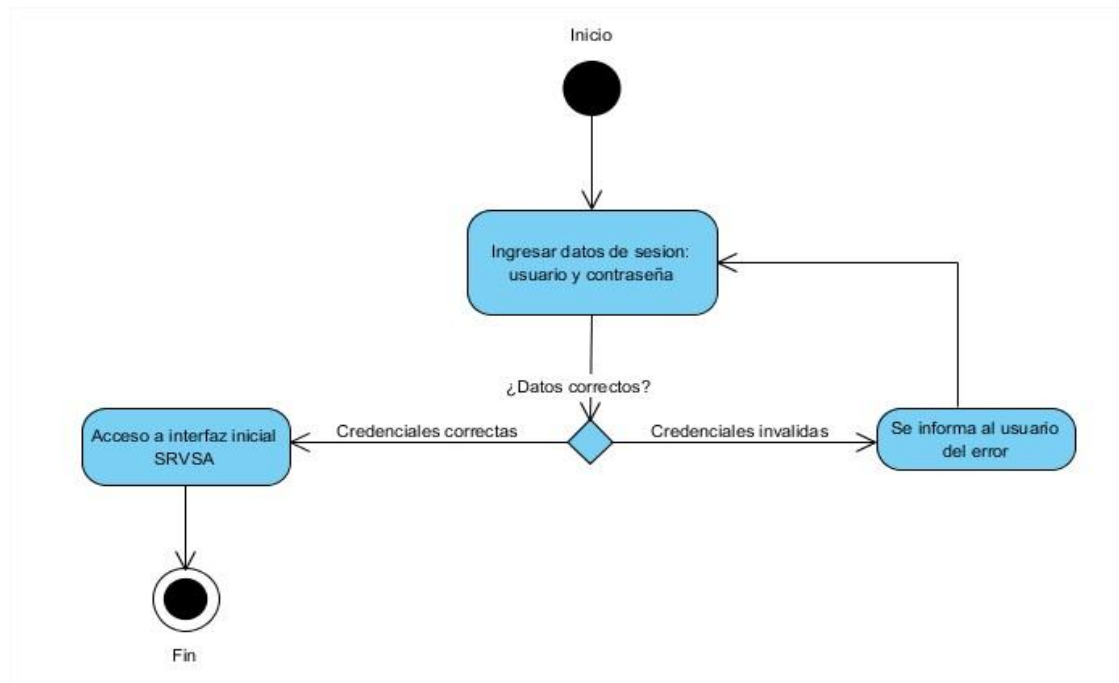


Ilustración 18: Diagrama de Actividad, ingreso al sistema



Administración de Catálogos

Se brinda el mantenimiento de los catálogos existentes en el sistema.

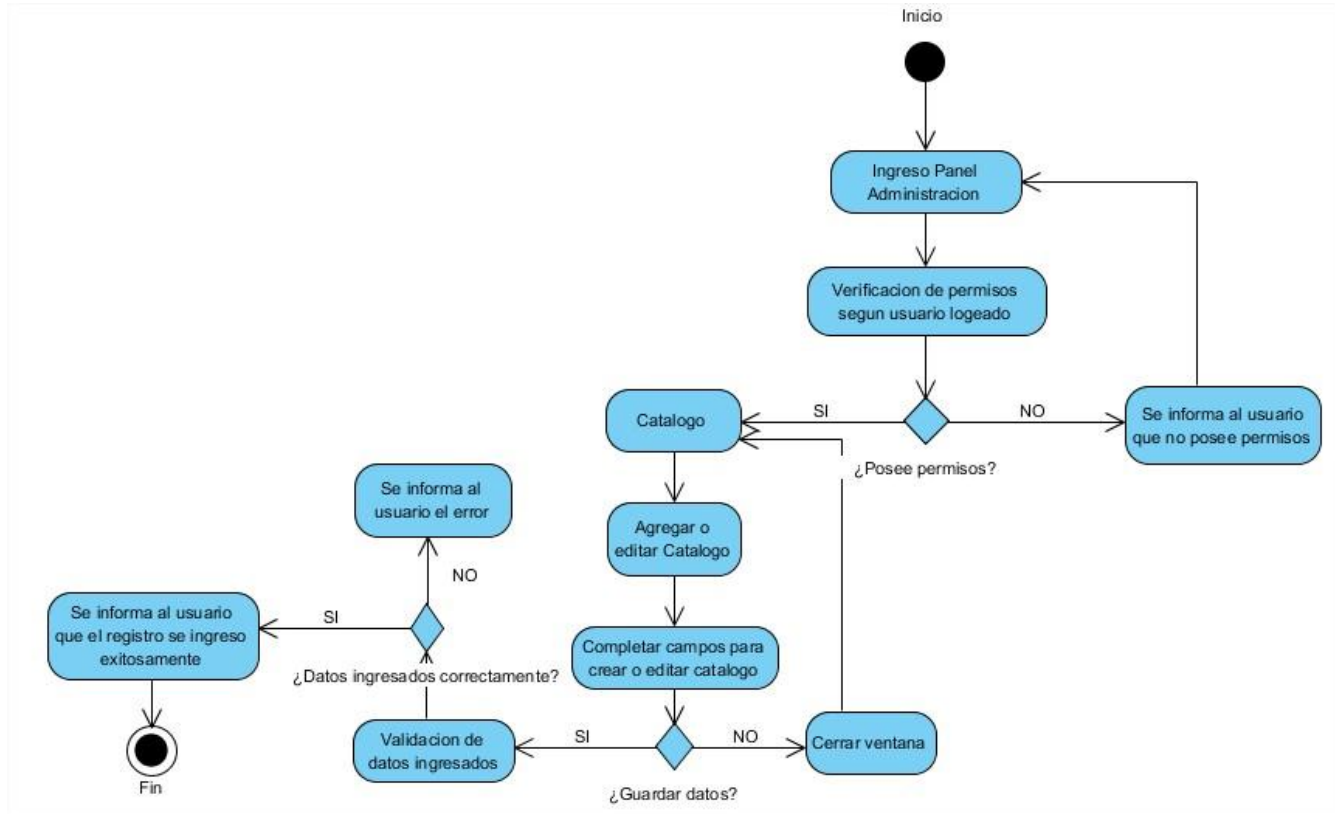


Ilustración 19: Diagrama de Actividad, administración de catálogos

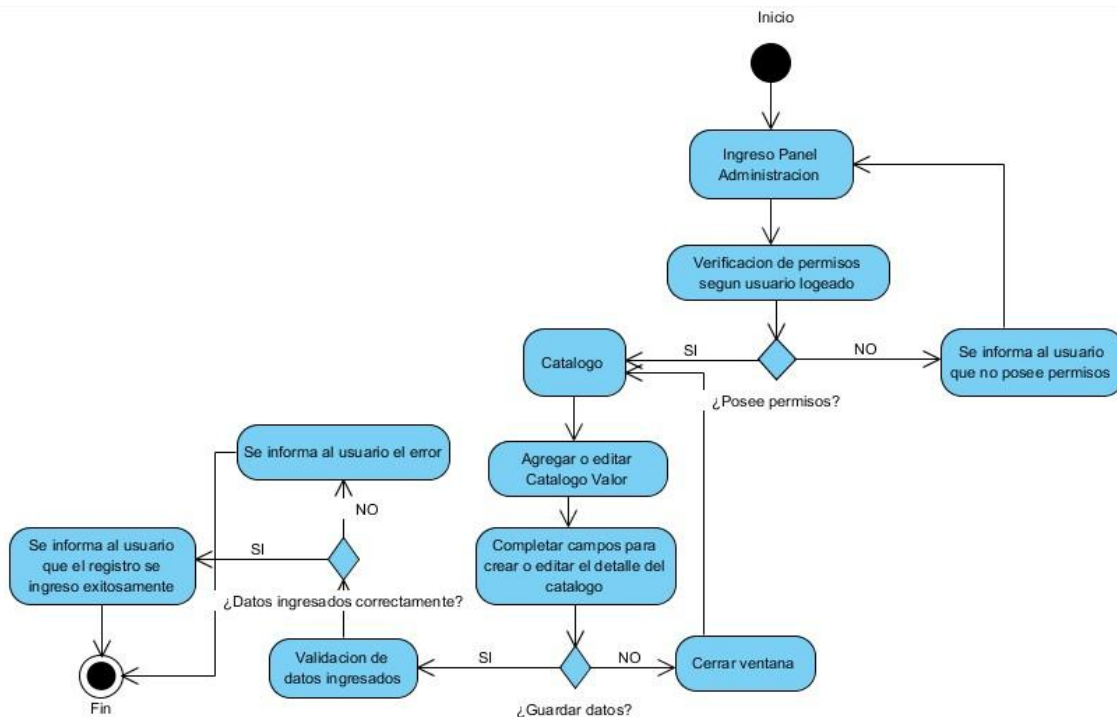


Ilustración 20: Diagrama de Actividad, administración de catálogo valor



Administración de Entrada

Este proceso se realiza para registrar la solicitud de abastecimiento de producto, de manera que cuando el producto se encuentre ingresado en bodega se aplica la entrada que posteriormente actualiza las existencias.

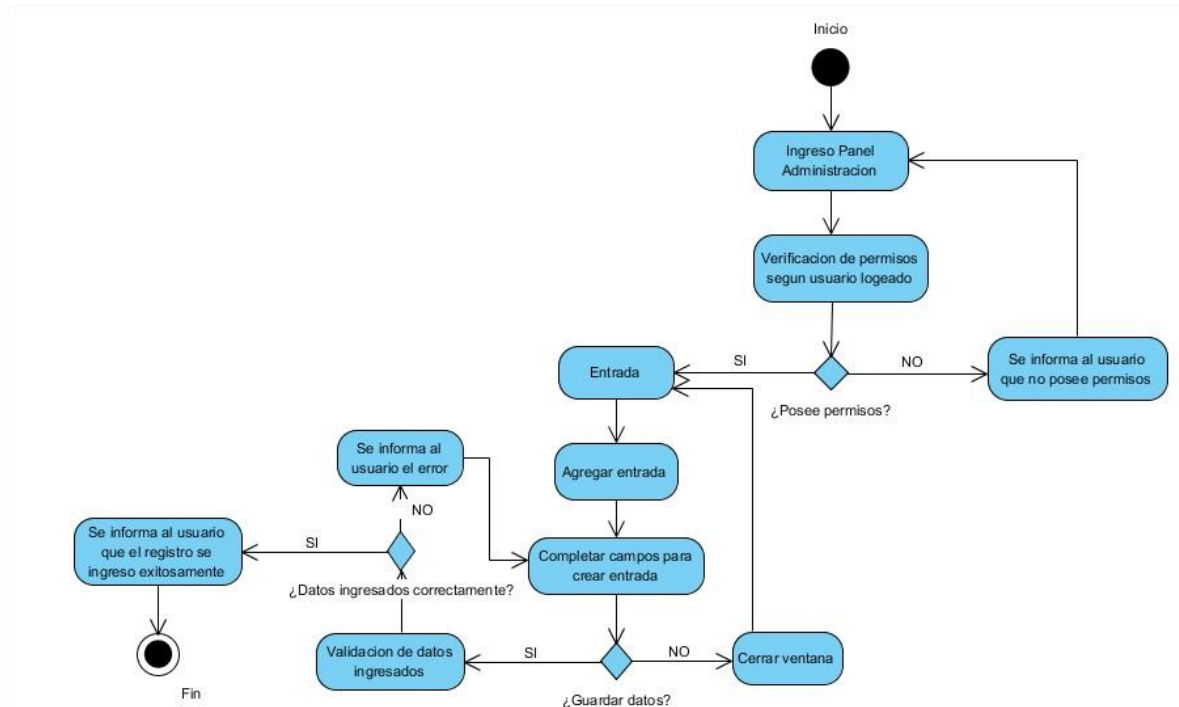


Ilustración 21: Diagrama de Actividad, administración de entrada

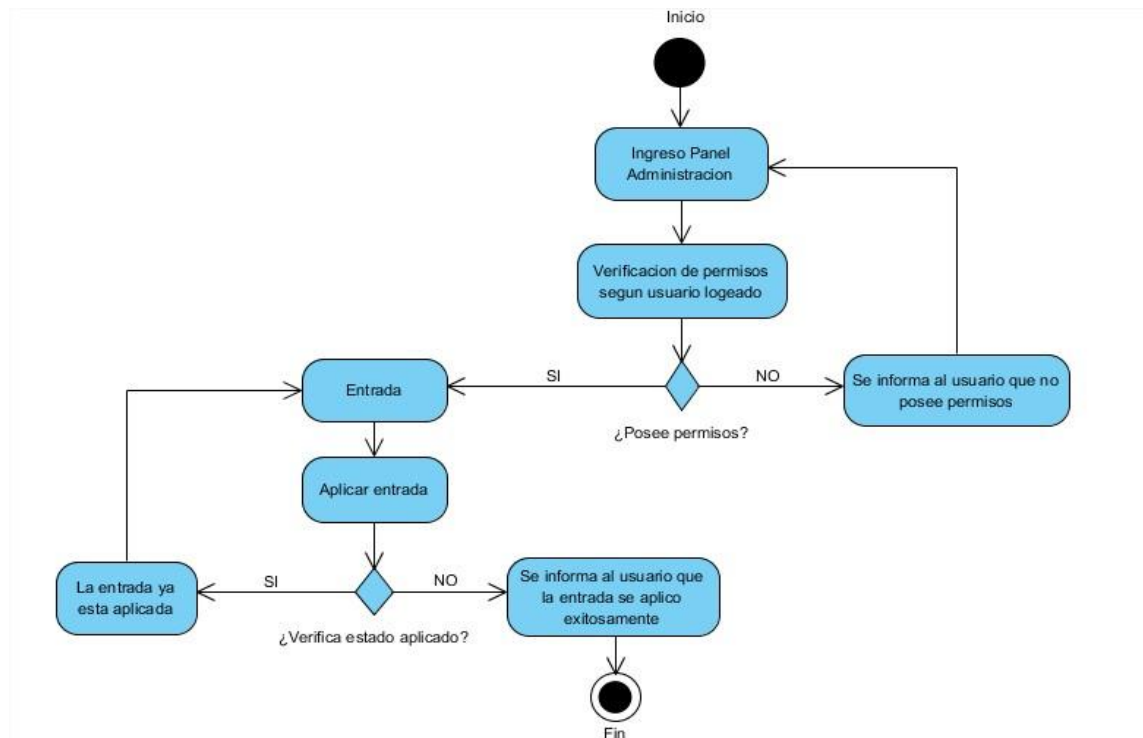


Ilustración 22: Diagrama de Actividad, Aplicar entrada



Búsqueda de piezas por categoría

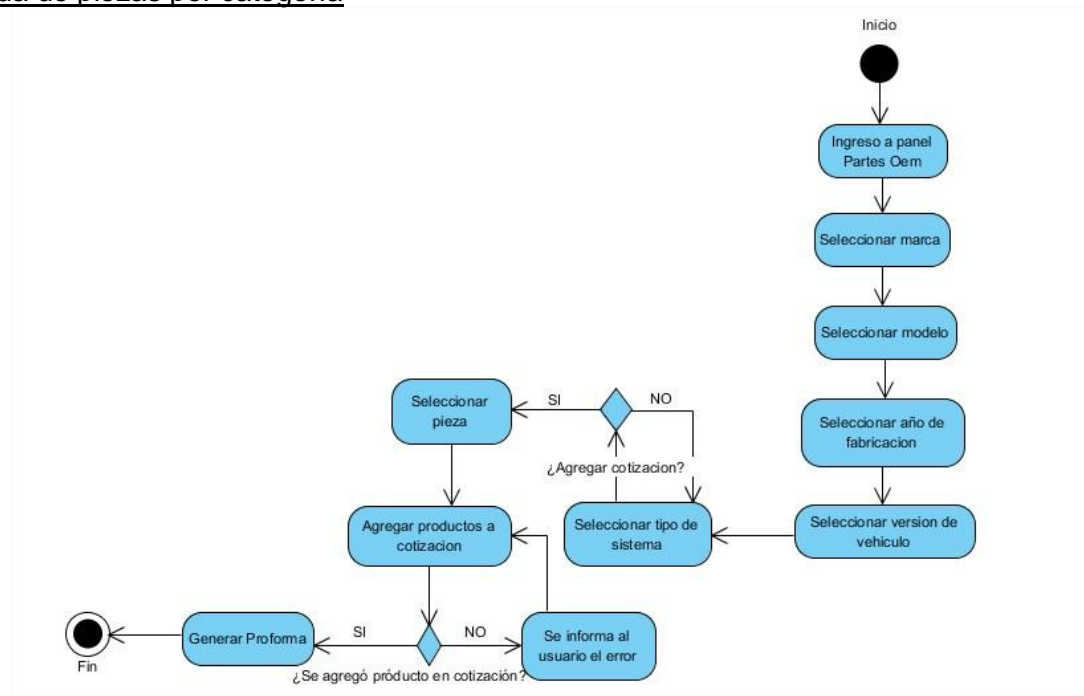


Ilustración 23: Diagrama de Actividad, Búsqueda de pieza por categoría usuario final

Búsqueda de piezas por producto

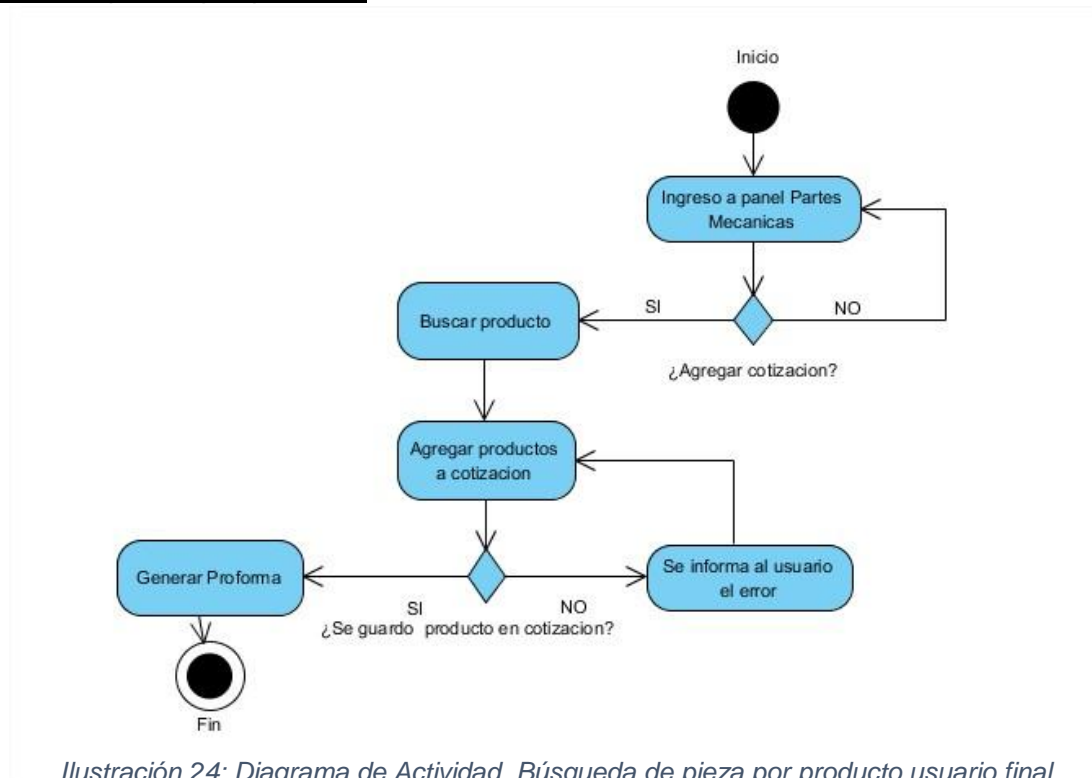


Ilustración 24: Diagrama de Actividad, Búsqueda de pieza por producto usuario final



Modelo de navegación

Este modelo presenta la estructura web del prototipo de sistema.

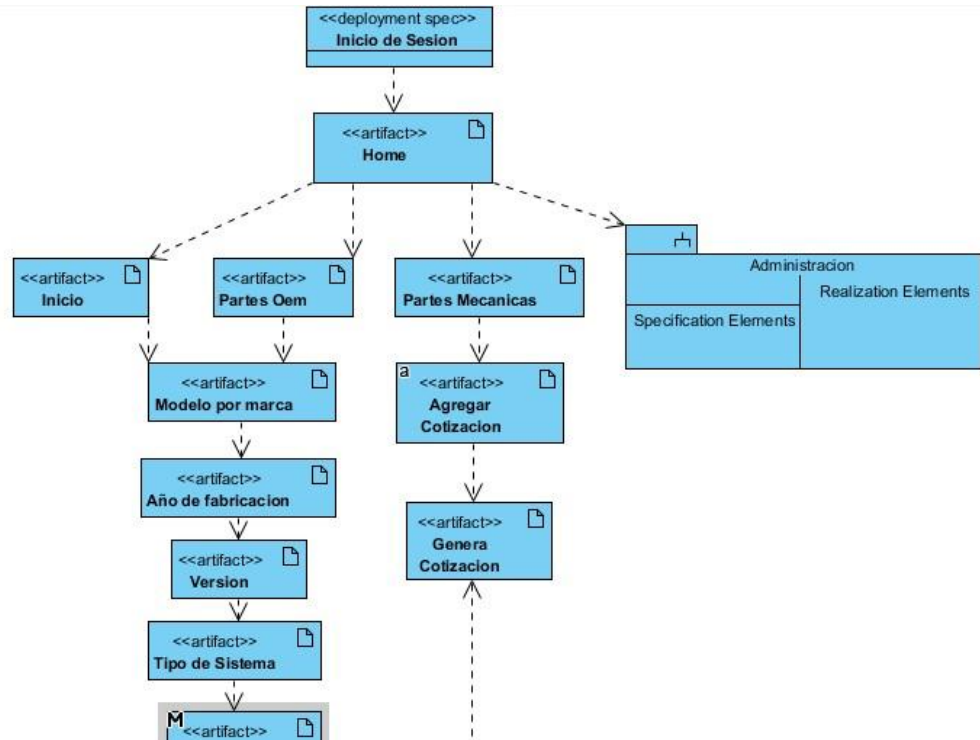


Ilustración 25: Modelo de Navegación SRVSA

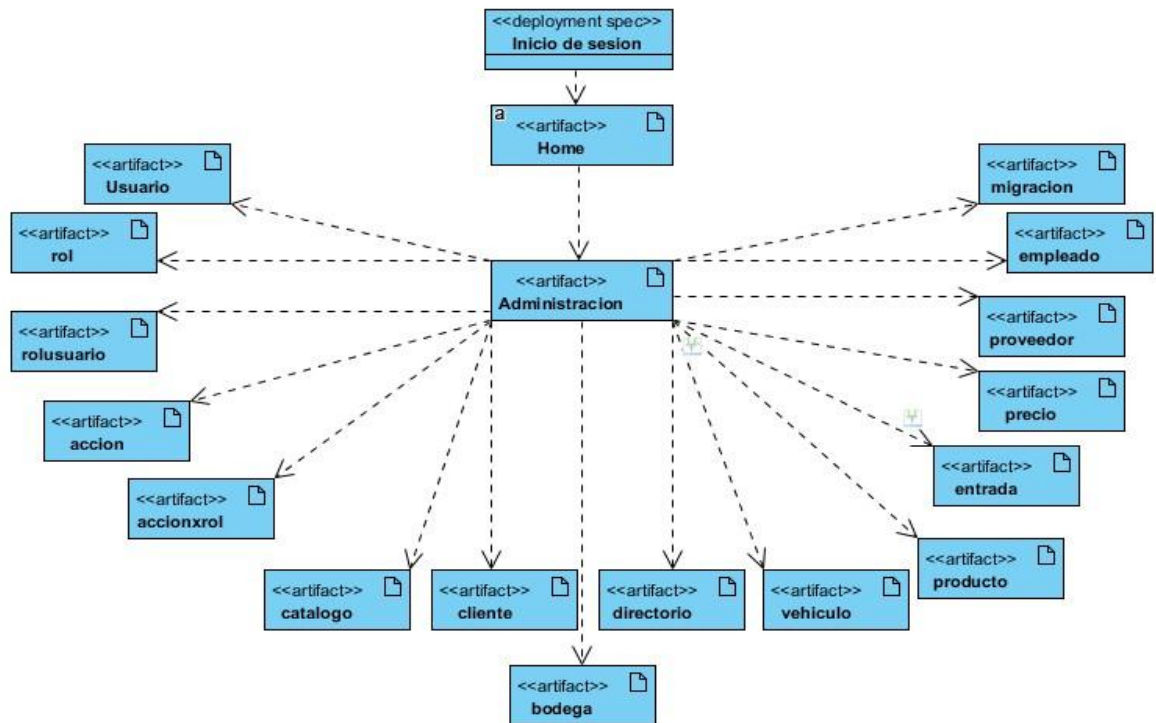


Ilustración 26: Modelo de Navegación de Administración SRVSA



Fase de construcción

En el siguiente diagrama se presentan los paquetes que componen el prototipo de sistema SRVSA.

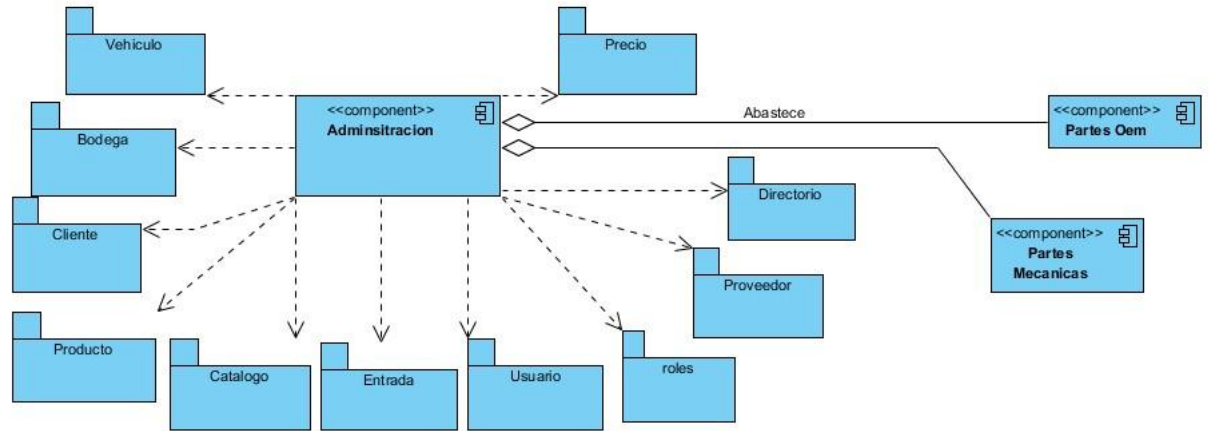


Ilustración 27: Diagrama de paquetes SRVSA

Patrón de arquitectura de desarrollo MVC

Es un patrón de diseño, desarrollo y arquitectura de software que separa la vista de la lógica del negocio a través de un módulo de comunicación bidireccional. Para cumplir con la estructura, el patrón propone tres componentes: modelo, vista y controlador, esto quiere decir que por una parte se define la estructura y presentación de la información y por el otro la interacción con el usuario.

El patrón utiliza dos conceptos de las Ciencias de la Computación, re-utilización de código y separación de conceptos características del patrón que buscan facilitar el desarrollo y la mantenibilidad del software.

Descripción

Modelo: Contiene el núcleo de la funcionalidad (dominio) de la aplicación. Encapsula el estado de la aplicación. Es independiente del Controlador y la Vista.

Vista: Es la presentación del Modelo. Puede acceder al Modelo, pero nunca cambiar su estado. Puede ser notificada cuando hay un cambio de estado en el Modelo.

Controlador: Reacciona a la petición del Cliente, ejecutando la acción adecuada y creando el modelo pertinente.

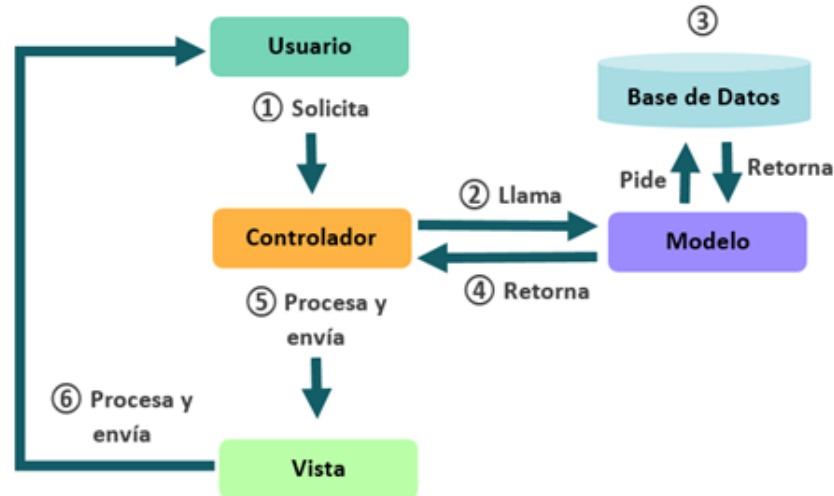


Ilustración 28: Arquitectura MVC

En esta ilustración se representa con flechas los modos de colaboración entre los distintos elementos que formarían una aplicación MVC. Los controladores, con su lógica de negocio, hacen de puente entre los modelos y las vistas. Pero además en algunos casos los modelos pueden enviar datos a las vistas. El flujo de trabajo característico en un esquema MVC sería el siguiente:

1. El usuario realiza una solicitud al sitio web. Generalmente estará desencadenada por acceder a una página del sitio. Esa solicitud le llega al controlador.
2. El controlador comunica tanto con modelos como con vistas. A los modelos les solicita datos o les manda realizar actualizaciones de los datos. A las vistas les solicita la salida correspondiente, una vez se hayan realizado las operaciones pertinentes según la lógica del negocio.
3. Para producir la salida, en ocasiones las vistas pueden solicitar más información a los modelos. En ocasiones, el controlador será el responsable de solicitar todos los datos a los modelos y de enviarlos a las vistas, haciendo de puente entre unos y otros. Sería corriente tanto una cosa como la otra, todo depende de la implementación; por eso esa flecha se presenta en otro color.
4. Las vistas envían al usuario la salida. Aunque en ocasiones esa salida puede ir de vuelta al controlador y sería éste el que hace el envío al cliente.

Ventajas del modelo

- Aplicaciones más robustas.



- Reutilización de los componentes. Es mucho más sencillo agregar múltiples representaciones de los mismos datos o información.
- Simplicidad en el mantenimiento de los sistemas.
- Desarrollo paralelo. Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de las otras capas.
- División de la lógica del negocio del diseño, aumentando la escalabilidad del proyecto.

Uso de MVC

Modelo

Representado por clases llamadas entidades, dichas entidades se encuentran almacenadas en la carpeta de **Models** del proyecto de MVC, cada entidad posee las propiedades heredadas desde la base de datos, el cual permite acceder a las tablas de la base de datos desde la entidad principal que contiene la conexión a la base datos.

A través del modelo se definen las relaciones, las entidades, vistas y/o procedimientos almacenados, propiedades, que podrán ser tratados por el usuario, entre otros.

```
namespace SRVSA.Models
{
    using System;
    using System.Data.Entity;
    using System.Data.Entity.Infrastructure;
    using System.Data.Entity.Core.Objects;
    using System.Linq;

    42 referencias | Auxiliadora Torrez, Hace 7 días | 1 autor, 11 cambios
    public partial class Entities : DbContext
    {
        20 referencias | Auxiliadora Torrez, Hace 35 días | 1 autor, 3 cambios | 0 excepciones
        public Entities()
            : base("name=Entities")
        {
        }

        0 referencias | Auxiliadora Torrez, Hace 35 días | 1 autor, 3 cambios | 0 excepciones
        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
            throw new UnintentionalCodeFirstException();
        }

        0 referencias | Auxiliadora Torrez, Hace 35 días | 1 autor, 3 cambios | 0 excepciones
        public virtual DbSet<accion> accion { get; set; }
        0 referencias | Auxiliadora Torrez, Hace 35 días | 1 autor, 3 cambios | 0 excepciones
        public virtual DbSet<accionxrol> accionxrol { get; set; }
        6 referencias | Auxiliadora Torrez, Hace 13 días | 1 autor, 1 cambio | 0 excepciones
        public virtual DbSet<bodega> bodega { get; set; }
        39 referencias | Auxiliadora Torrez, Hace 35 días | 1 autor, 3 cambios | 0 excepciones
        public virtual DbSet<catalogo> catalogo { get; set; }
        76 referencias | Auxiliadora Torrez, Hace 35 días | 1 autor, 3 cambios | 0 excepciones
        public virtual DbSet<catalogovalor> catalogovalor { get; set; }
        3 referencias | Auxiliadora Torrez, Hace 34 días | 1 autor, 1 cambio | 0 excepciones
        public virtual DbSet<cliente> cliente { get; set; }
        10 referencias | Auxiliadora Torrez, Hace 19 días | 1 autor, 1 cambio | 0 excepciones
        public virtual DbSet<directorio> directorio { get; set; }
    }
}
```

Ilustración 29: Diagrama de Clase SRVSA.Models



Las entidades de utilerías para reportes u otros usos, se encuentran en la carpeta de Utilidades del proyecto de MVC, dichas entidades interactúan con entidades que se encuentran en el modelo principal para el acceso a datos.

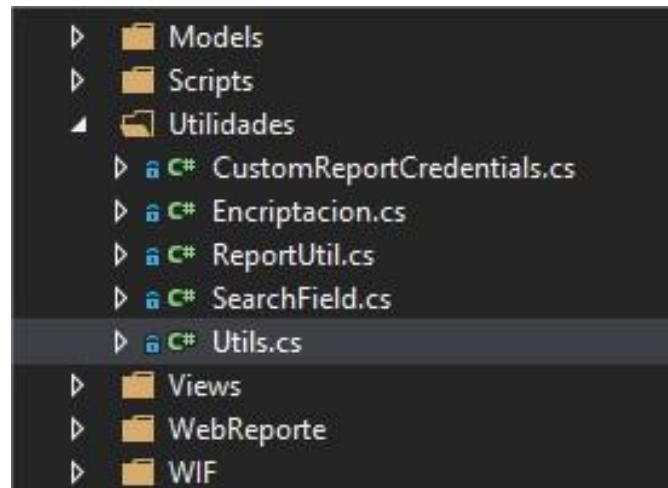


Ilustración 30: Diagrama de Acceso a entidades de utilidades de SRVSA

```
namespace SRVSA.Utilidades
{
    1 referencia | J.Laguna, Hace 4 días | 1 autor, 2 cambios
    public static class ReportUtil
    {
        0 referencias | J.Laguna, Hace 4 días | 1 autor, 1 cambio | 0 excepciones
        public static string GetUrlReport(this UrlHelper controller, int IdReporte, bool ShowParamPromts = false)
        {
            string UrlVisor = controller.Content("~/WebReporte/WebVisorSSRS.aspx");

            UrlVisor += "?pReporte=" + Encriptacion.EncriptarBase64(CreateParamReport(IdReporte, null, ShowParamPromts));

            return UrlVisor;
        }

        0 referencias | J.Laguna, Hace 4 días | 1 autor, 1 cambio | 0 excepciones
        public static string GetUrlReport(this UrlHelper controller, int IdReporte, object Values, bool ShowParamPromts = false)
        {
            string UrlVisor = controller.Content("~/WebReporte/WebVisorSSRS.aspx");

            var param = new Dictionary<string, object>();

            if (Values != null)
            {
                var lsProperties = TypeDescriptor.GetProperties(Values, false).OfType<PropertyDescriptor>().ToList();
                lsProperties.ForEach(c => { param.Add(c.Name, c.GetValue(Values)); });
            }

            UrlVisor += "?pReporte=" + Encriptacion.EncriptarBase64(CreateParamReport(IdReporte, param, ShowParamPromts));

            return UrlVisor;
        }
    }
}
```

Ilustración 31: Diagrama de Clase ReportUtil para configuración del visor de reportes



Las entidades utilizadas para la administración de la seguridad de SRVSA, se encuentran en la carpeta de **WIF** del proyecto de MVC, dichas entidades interactúan con las entidades de seguridad que se encuentran en el modelo principal para el acceso a datos.

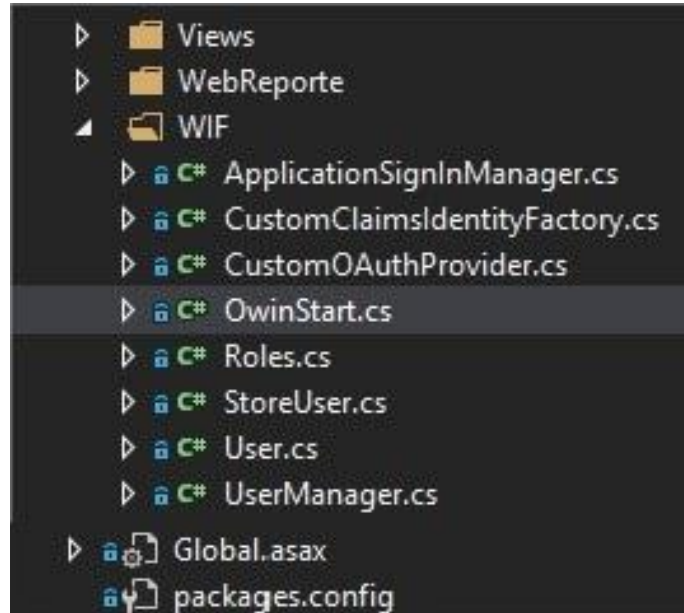


Ilustración 32: Diagrama de Acceso a entidades de seguridad de SRVSA

Controlador

Son las clases que enlazan los datos procesados, dichos datos son obtenidos desde la lógica del negocio, que posteriormente son enviados a la vista para la visualización del lado del cliente. Existen controladores bases, los cuales controlan el funcionamiento de las vistas según el sector que se trabaje:

1. Para inicio de sesión **Controllers/LoginController.cs**,
2. Para inicio del sistema **Controllers/HomeController.cs**,
3. Para búsqueda de partes por código categoría
Controllers/PartoemController.cs,
4. Para búsqueda de piezas ya sea por nombre o por códigos
Controllers/PartMecanciaController.cs.



Para la administración de SRVSA, cada sección posee su propio controlador:

1. Producto: **Controllers/ProductoController.cs**
2. Catálogo: **Controllers/CatalogoController.cs**
3. Cliente: **Controllers/ClienteController.cs**
4. Directorio: **Controllers/DirectorioController.cs**
5. Entrada: **Controllers/InventarioEntradaController.cs**
6. Empleado: **Controllers/EmpleadoController.cs**
7. Bodega: **Controllers/BodegaController.cs**
8. Precio: **Controllers/PrecioController.cs**
9. Proveedor: **Controllers/ProveedorController.cs**
10. Vehículo: **Controllers/VehiculoController.cs**
11. Reportes: **Controllers/ReporteController.cs**
12. Seguridad de usuarios: **Controllers/SeguridadController.cs**

```
namespace SRVSA.Controllers
{
    0 referencias | Auxiliadora Torrez, Hace 19 días | 1 autor, 3 cambios
    public class HomeController : Controller
    {
        Models.Entities ctx = new Models.Entities();
        // GET: Home
        0 referencias | Auxiliadora Torrez, Hace 98 días | 1 autor, 1 cambio | 0 solicitudes | 0 excepciones
        public ActionResult Index()
        {
            List<SelectListItem> items = new List<SelectListItem>();

            var items1 = (from marca in ctx.catalogovalor
                          join cat in ctx.catalogo
                          on marca.IdCatalogo equals cat.IdCatalogo
                          where cat.Codigo == "MARCAV"
                          orderby marca.Valor ascending
                          select new
                          {
                              Codigo = marca.IdCatalogoValor,
                              Text = marca.Valor
                          }).ToList();

            ViewBag.Marca = new SelectList(items1, "Codigo", "Text");

            var tipo = (from marca in ctx.catalogovalor
                        join cat in ctx.catalogo
                        on marca.IdCatalogo equals cat.IdCatalogo
                        where cat.Codigo == "TIPOV"
                        select new
                        {
                            Codigo = marca.IdCatalogoValor,
                            Text = marca.Valor
                        });
        }
    }
}
```

Ilustración 33: Diagrama de Controlador Home SRVSA



```
namespace SRVSA.Controllers
{
    0 referencias | Auxiliadora Torrez, Hace 10 días | 1 autor, 5 cambios
    public class PartoemController : Controller
    {
        Models.Entities ctx = new Models.Entities();

        #region "Vistas"
        // GET: Partoem
        0 referencias | Auxiliadora Torrez, Hace 35 días | 1 autor, 2 cambios | 0 solicitudes | 0 excepciones
        public ActionResult Index()
        {
            return View();
        }

        0 referencias | Auxiliadora Torrez, Hace 19 días | 1 autor, 3 cambios | 0 solicitudes | 0 excepciones
        public ActionResult Modelo(int Id)
        {
            var directm = (from marca in ctx.directorio
                           join catv in ctx.catalogovalor
                               on marca.IdCatalogoValor equals catv.IdCatalogoValor
                           join modelo in ctx.catalogovalor
                               on catv.IdCatalogoValorAnterior equals modelo.IdCatalogoValor
                           where marca.Activo == true && catv.IdCatalogoValorAnterior == Id
                           select new ClsModelo
                           {
                               DirModelo = marca.IdDirectorio,
                               modelo = modelo.Valor.ToLower(),
                               marca = catv.Valor.ToLower(),
                               IdMarca = catv.IdCatalogoValor
                           }).ToList();

            return View(directm);
        }
    }
}
```

Ilustración 34: Diagrama de Controlador Partoem SRVSA

Vista

En la aplicación son representadas por archivo cshtml donde se envían la salida HTML que recibirá el cliente.

```
@model SRVSA.Models.ClsMarca

@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
    @section JavaScript
    {
        <script type="text/javascript">

            $(document).ready(function () {

                $.getJSON("../Home/GetRutasImagenes", function (data) {

                    console.log(data);

                    var lsImg = [];

                    var divMarcas = $('#divMarcas');

                    var html = " <div class='service green hs-block' data-rotate-x='50deg' data-move-z='-50px' data-move-y='50px'> ";
                    for (var i = 0; i < data.urls.length; i++) {
                        if (i > 0) {
                            html += " <div class='service green hs-block' data-rotate-x='50deg' data-move-z='-50px' data-move-y='50px'> ";
                        }

                        html += "<div class='icon'> ";
                        html += "<a>";

                        var ImgId = data.urls[i].IdMarca.toString();
                        var ImgOver = data.url[i].DirMarca;
                        lsImg.push(ImgId);
                    }
                });
            });
        </script>
    }
}
```

Ilustración 35: Diagrama de Vista Home SRVSA



Presentación de piezas modeladas en ambiente web

Para mostrar las piezas modelada en ambiente web son necesarios cumplir con ciertas condiciones previas:

1. Se deben almacenar las librerías de **three.js** necesarias para la carga de los modelos 3D.
2. Los modelos 3D deben ser exportados a los formatos adecuados que soporta Three.js, en nuestro caso particular se en formato OBJ con su respectivo MTL.
3. Los modelos 3D en formato OBJ deberán ser colocados en el mismo directorio que sus respectivos MTL e imágenes de textura.

Para la interacción del modelo 3D con las vistas del sistema se crean vistas parciales en la cuales se hace uso de la **librería three.js**. para cargar el modelo 3D que corresponde con la pieza, esto es posible gracias a que a través del controlador a la vista parcial se transfiere el valor del código OEM que le corresponde a la pieza de modo que, los modelos se muestran en dependencia de la pieza que se desee ver.

```
PartMecanicaController.cs* X
SRVSA
SRVSA.Controllers.PartMecanicaController

5  using System.Web;
6  using System.Web.Mvc;
7
8  namespace SRVSA.Controllers
9  {
10     0 referencias | Auxiliadora Torrez, Hace 8 días | 1 autor, 6 cambios
    public class PartMecanicaController : Controller
11     {
12         Models.Entities ctx = new Models.Entities();
13
14         // GET: PartMecanica
15         0 referencias | Auxiliadora Torrez, Hace 127 días | 1 autor, 1 cambio | 0 solicitudes | 0 excepciones
        public ActionResult Index()...
16
17
18         0 referencias | Auxiliadora Torrez, Hace 8 días | 1 autor, 1 cambio | 0 solicitudes | 0 excepciones
19         public ActionResult _DetallePartes(string Id)
20         {
21             ViewBag.codigo= Id;
22             return View();
23         }
24     }
25 }
```

Ilustración 36: Transferencia de código OEM Vista-Controlador



```
10 </head>
11 </head>
12 <body>
13 <script src="../../Scripts/three.min.js"></script>
14 <script src="../../Scripts/js/controls/TrackballControls.js"></script>
15 <script src="../../Scripts/js/Detector.js"></script>
16 <script src="../../Scripts/loaders/DOSLoader.js"></script>
17 <script src="../../Scripts/loaders/MTLLoader.js"></script>
18 <script src="../../Scripts/loaders/OBJLoader.js"></script>
19 <script src="../../Scripts/stats.min.js"></script>
20
21 <script type="text/javascript">
22
23     if (!Detector.webgl) Detector.addGetWebGLMessage();
24
25     var STATS_ENABLED = false;
26
27     var CAR = "Html.Raw(ViewBag.codigo)";
28
29     var container, stats;
30     var camera, controls, scene, renderer;
31     var pickingData = [], pickingTexture, pickingScene;
32     var objects = [];
33     var highlightBox;
34
35     var mouse = new THREE.Vector2();
36     var offset = new THREE.Vector3(10, 10, 10);
37
38     init();
39     animate();
40
41     function init() {
42
43         camera = new THREE.PerspectiveCamera(70, window.innerWidth / window.innerHeight, 1, 10000);
44         camera.position.z = 1000;
```

Ilustración 37: Vista parcial _DetallePartes

```
151     scene.add(highlightBox);
152
153     // model
154
155     var onProgress = function (xhr) {
156         if (xhr.lengthComputable) {
157             var percentComplete = xhr.loaded / xhr.total * 100;
158             console.log(Math.round(percentComplete, 2) + '% downloaded');
159         }
160     };
161
162     var onError = function (xhr) { };
163
164     THREE.Loader.Handlers.add(/\.dds$/i, new THREE.DDSLoader());
165
166     var mtlLoader = new THREE.MTLLoader();
167     mtlLoader.setPath('../modelo/');
168     mtlLoader.load(CAR + '.mtl', function (materials) {
169
170         materials.preload();
171
172         var objLoader = new THREE.OBJLoader();
173         objLoader.setMaterials(materials);
174         objLoader.setPath('../modelo/');
175         objLoader.load(CAR + '.obj', function (object) {
176
177             object.position.set(0, 0, 0);
178             object.scale.x = object.scale.y = object.scale.z = 100;
179             scene.add(object);
180
181         }, onProgress, onError);
182     });
183
184
```

Ilustración 38: Llamado de modelo 3D



Para la interacción del cliente con la información que se encuentre en la vista y el modelo 3D el llamado de la vista parcial que posee la carga del modelo 3D se debe realizar a través **IFRAME** que nos permite manipular y combinar dos vistas independientes en una sola vista de tal manera que la vista parcial se maneja como si fuera una vista principal.

```
107     var iframe = document.createElement('iframe');
108     iframe.height = '414';
109     iframe.width = '550';
110     iframe.style = 'border:none;';
111     iframe.src = '/PartMecanica/ DetallePartes?Id=' + Id;
112     document.getElementById("divModelo").innerHTML = "";
113     document.getElementById('divModelo').appendChild(iframe);
114
```

Ilustración 39: Incrustación de vista de modelo 3D en vista principal

Fase de cierre

En esta Fase la aplicación recientemente desarrollada reemplazará a la anterior. Mientras está ejecutándose en paralelo con la aplicación anterior, la nueva se prueba, los usuarios son entrenados y los procedimientos de la organización se cambian antes de que ocurra el cierre.

Implementación de la aplicación

Entornos de desarrollo

El SRVSA fue desarrollado en el sistema operativo Windows 10 con el lenguaje de programación C# con el patrón de diseño web MVC, y el framework Entity Framework como herramienta de mapeo objeto-relacional. Se utilizó el IDE Visual Studio 2015. La base de datos utilizada fue SQL Server 2014. En la interfaz del usuario se utilizó Bootstrap dado a su facilidad en la creación de diseños web, con el apoyo de JQuery para personalizar los efectos, JQueryDatatable para las tablas. Y se utilizó el lenguaje JavaScript y Ajax para tareas como la validación de formularios y comunicación vista controlador para la transferencia de datos al controlador.

Estas tecnologías fueron seleccionadas debido a que estas no tienen ningún costo, y cuentan con una amplia gama de documentación en internet, ya que son muy utilizadas en el desarrollo de aplicaciones web.



Pruebas del sistema

Casos de Prueba

Se documentaron los casos de prueba principales para asegurar las funcionalidades esenciales del sistema. Esto permitió evaluar y verificar que el software cumple con los requerimientos funcionales, determinando así, que los resultados de las pruebas fueron en la mayoría de los casos satisfactorios

Prueba 1 Carga de modelo 3D en sistema	
Objetivo de la prueba	Cargar modelos 3D de las piezas
Precondición	Buscar una pieza en el catálogo del sistema
Descripción de la prueba	Seleccionar el código OEM de la pieza del catálogo del sistema
resultados esperados	Al seleccionar el código OEM de la Pieza se mostrará el modelo 3D con texturas realistas de la pieza correspondiente
Resultados obtenidos	La pieza de carga sin ningún tipo de textura
Estado	No satisfactorio

Tabla 27: Carga de modelos 3D en sistema

Prueba 2 Manipulación del contenido de la pagina	
Objetivo de la prueba	hacer uso de las barras de búsqueda mientras el modelo 3D este cargado
Precondición	haber seleccionado una pieza y cargado su respectivo modelo 3D
Descripción de la prueba	mientras se muestra el modelo 3D en pantalla se hará uso de los componentes de la pagina
resultados esperados	poder hacer búsquedas de otras piezas vehiculares y poder agregar la respectiva cotización
Resultados obtenidos	la librería three.js inhabilita todo el contenido que se encuentra en la página principal a excepción del modelo 3D dificultando la manipulación de los datos existentes en la vista.
Estado	No satisfactorio

Tabla 28: Manipulación de objetos 3D y contenido Web



Prueba 3 Registrar catálogos	
Objetivo de la prueba	El usuario administrador registrara un catalogo
Precondición	Iniciar sesión en el sistema
Descripción de la prueba	Registrar un nuevo catálogo con descripción: tipo combustión del vehículo con sus valores correspondientes de gasolina y diésel
Resultados esperados	El sistema muestra un mensaje notificando que el catalogo se creó exitosamente.
Resultados obtenidos	El catalogo se creó exitosamente con sus valores correspondiente.
Estado	satisfactorio

Tabla 29: registrar un nuevo catalogo

Prueba 4 Registrar vehículo	
Objetivo de la prueba	El usuario administrador registrara un vehículo
Precondición	Iniciar sesión en el sistema
Descripción de la prueba	Registrar un nuevo vehículo con marca: Hyundai modelo: veloster , versión: turbo , tipo de combustión: gasolina , cilindraje: L4 , capacidad: 1526 y año: 2015
resultados esperados	El sistema muestra un mensaje notificando que el vehículo se creó exitosamente.
Resultados obtenidos	El vehículo se ingresó exitosamente.
Estado	satisfactorio

Tabla 30: registrar un nuevo vehículo

Prueba 5 Registrar productos	
Objetivo de la prueba	El usuario administrador registrara un producto
Precondición	Iniciar sesión en el sistema
Descripción de la prueba	Registrar un nuevo producto con nombre: tensor , descripción: tensor , con código OEM: 2441026000 , tipo de sistema: motor , vehículo compatible: Hyundai Excel 1989
resultados esperados	El sistema muestra un mensaje notificando que el producto se creó exitosamente.
Resultados obtenidos	El producto se ingresó exitosamente.
Estado	satisfactorio

Tabla 31: registrar nuevo producto



Prueba 6 Registrar entrada	
Objetivo de la prueba	El usuario administrador registrara una solicitud de entrada
Precondición	Iniciar sesión en el sistema
Descripción de la prueba	Registrar una solicitud de entrada con bodega: bodega1 , concepto: concepto por compra , con detalle de entrada ubicación: localizacion2 , producto: rodamiento delantero , cantidad: 3 . ubicación: exhibición , producto: cubierta delantera superior , cantidad: 6 . Aplicar entrada
resultados esperados	El sistema muestra un mensaje notificando que la solicitud de entrada se creó exitosamente. Luego de haber sido creada la solicitud se aplica la entrada y el sistema muestra mensaje notificando que la entrada se aplicó exitosamente
Resultados obtenidos	La entrada se ingresó exitosamente.
Estado	satisfactorio

Tabla 32: registrar entrada

Prueba 7 Registrar precio	
Objetivo de la prueba	El usuario administrador registrara un precio
Precondición	Iniciar sesión en el sistema
Descripción de la prueba	Registrar un precio con costo: 14.58 , precio mayor por unidad: 14.58 , precio de venta: 14.58 , moneda: córdoba
resultados esperados	El sistema muestra un mensaje notificando que el precio se creó exitosamente.
Resultados obtenidos	El precio se ingresó exitosamente.
Estado	satisfactorio

Tabla 33: registrar precio

Prueba 8 Registrar proveedor	
Objetivo de la prueba	El usuario administrador registrara un proveedor
Precondición	Iniciar sesión en el sistema
Descripción de la prueba	Registrar un precio con nombre: SABRITA , identificación: 142412 , dirección: GUADALAJARA
resultados esperados	El sistema muestra un mensaje notificando que el proveedor se creó exitosamente.
Resultados obtenidos	El proveedor se ingresó exitosamente.



Prueba 8 Registrar proveedor

Estado	satisfactorio
--------	---------------

Tabla 34: registrar proveedor

Prueba 9 Registrar bodega

Objetivo de la prueba	El usuario administrador registrara una bodega
Precondición	Iniciar sesión en el sistema
Descripción de la prueba	Registrar una bodega con nombre: BODEGA1 , dirección: BOACO
resultados esperados	La bodega se ingresó exitosamente.
Resultados obtenidos	La bodega se ingresó exitosamente.
Estado	satisfactorio

Tabla 35: registrar bodega

Prueba 10 Buscar pieza por categoría

Objetivo de la prueba	El usuario final buscara pieza por categoría
Precondición	Iniciar sesión en el sistema
Descripción de la prueba	Buscar pieza por categoría seleccionando marca Hyundai , modelo: Accent , año: 2008 , versión: L4 1.5 GAS GLS , tipo de sistema: alternador , seleccionar una imagen
resultados esperados	mostrar pieza 3D según la imagen seleccionada de la pieza
Resultados obtenidos	La pieza 3D se visualiza y se manipula según se desee
Estado	satisfactorio

Tabla 36: Buscar pieza por categoría

Soluciones a pruebas no satisfactorias

Prueba 1 Carga de modelo 3D en sistema

Objetivo de la prueba	Cargar modelos 3D de las piezas
Estado	No Satisfactorio
Error detectado	Los modelos 3D diseñados no cargaron ninguna de las texturas correspondientes, esto debido a que los materiales de las texturas eran de tipo MentalRay (Software para renderizar) y los materiales de ese tipo no son compatibles con el renderizado WebGL de three.js
Solución a Error	Los materiales de las texturas fueron cambiados a tipo Standard y se les dio las iluminaciones necesarias para su aspecto metálico

Tabla 37: Solución a prueba 1

Prueba 2 Manipulación del contenido de la página



Objetivo de la prueba	hacer uso de las barras de búsqueda mientras el modelo 3D este cargado
Estado	No Satisfactorio
Error detectado	la librería three.js inhabilita todo el contenido que se encuentra en la página principal a excepción del modelo 3D dificultando la manipulación de los datos existentes en la vista.
Solución a Error	se utilizó el IFRAME que es un documento HTML incrustado dentro de otro documento HTML, para ello el modelo 3D tuvo que ser cargado a través de una vista parcial, que nos permitió combinar las dos vistas para la carga de los modelos según la pieza que se desee ver.

Tabla 38: Solución a prueba 2

Primer prototipo de Serie SRVSA

Interfaz Principal del sistema



Ilustración 40: Inicio de Sesión SRVSA



Pestaña Inicio Superior

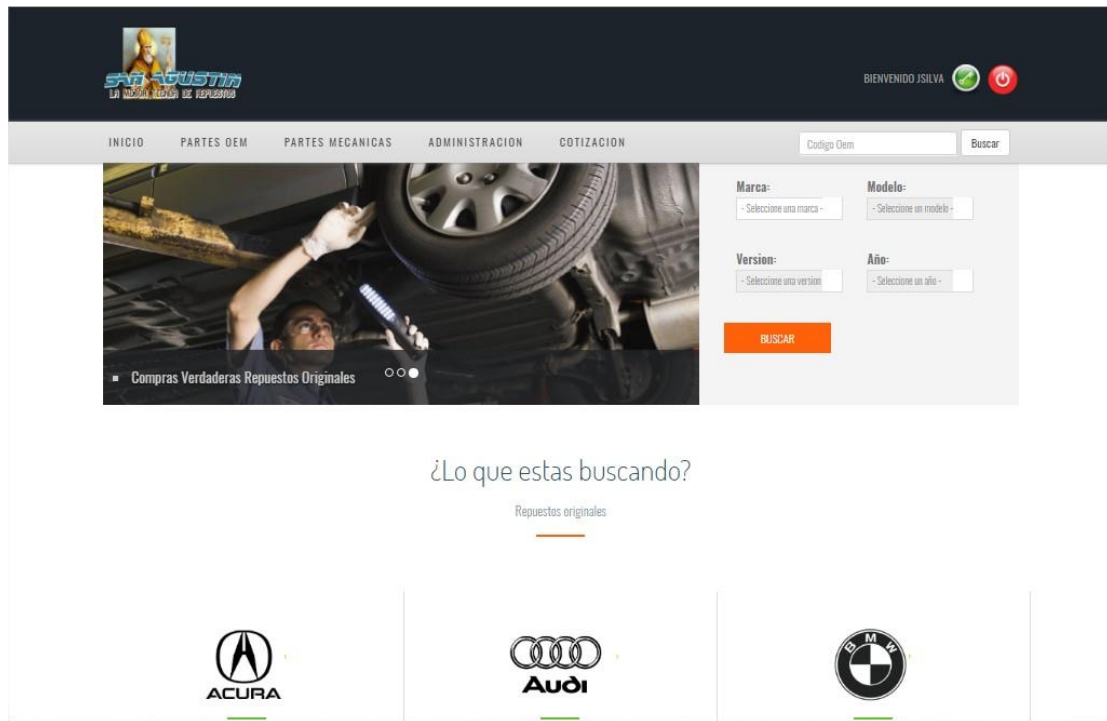


Ilustración 41: Pestaña Inicio Superior SRVSA

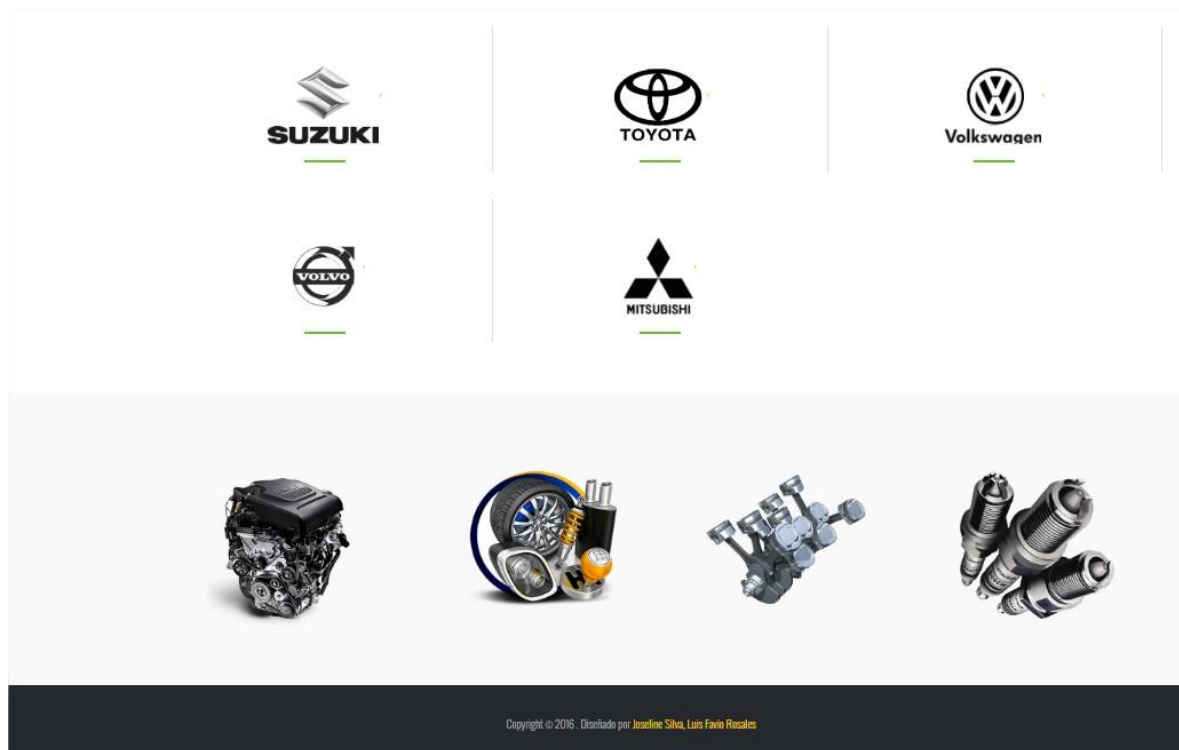


Ilustración 42: Pestaña Inicio Inferior SRVSA

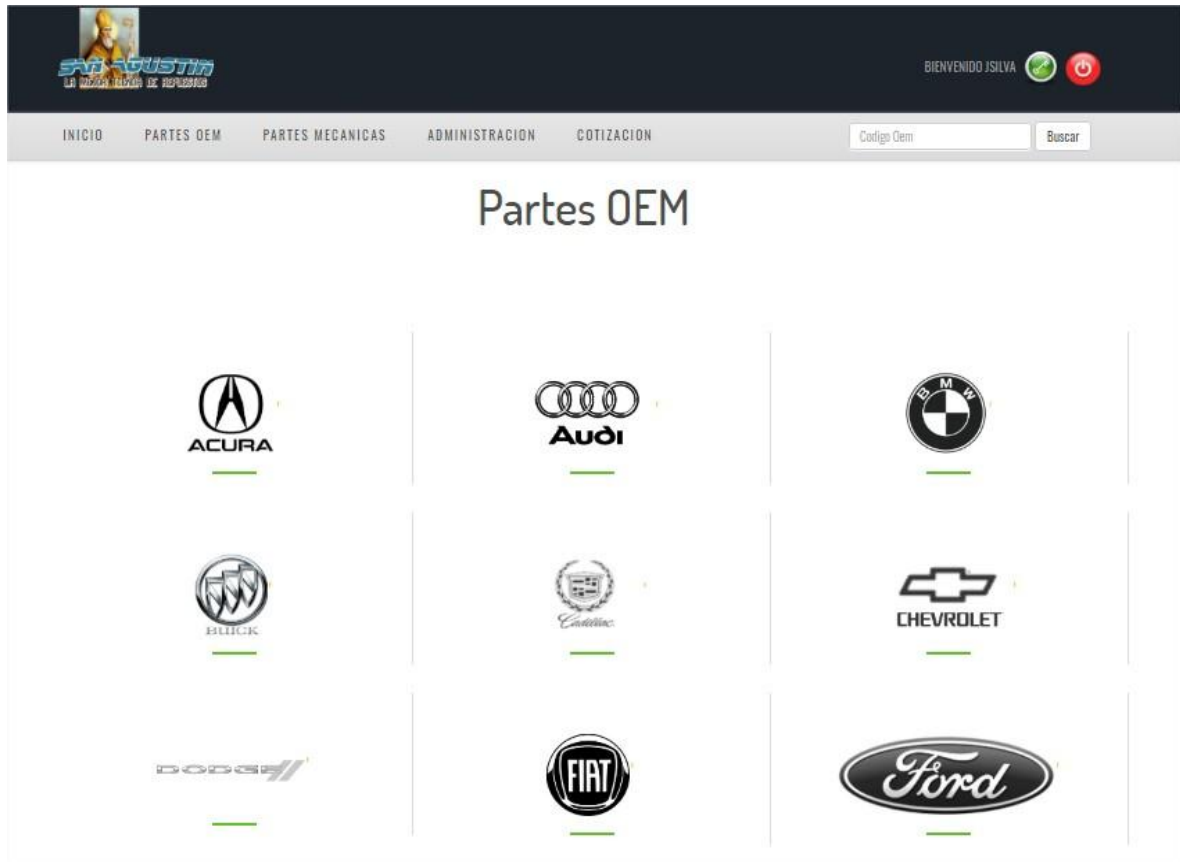


Ilustración 43: Pestaña Partes OEM SRVSA

BIENVENIDO JSILVA

INICIO

PARTES OEM

PARTES MECANICAS

ADMINISTRACION

COTIZACION

Codigo Oem

Buscar

Buscar:

Nombre	CodigoOem	SubCodigoOem	Marca
ALTERNADOR	3730023600		HYUNDAI
RODAMIENTO DELANTERO	373342B100		HYUNDAI
RODAMIENTO POSTERIOR	3734237400		HYUNDAI
CASE DELANTERO	3733022651		HYUNDAI
CASE POSTERIOR	3736022651		HYUNDAI
ROTOR	3734022650		HYUNDAI
STATOR	3735022650		HYUNDAI
REGULADOR DE VOLTAJE	3737022650		HYUNDAI
CABLE DE CONTROL DE CAMBIO	467601G100		HYUNDAI
EJE TRANSVERSAL	4500022BC6		HYUNDAI

ROTOR
3734022650

+ Agregar Cotizacion

Agregar al carrito

Generar Cotizacion

Mostrando registros del 1 al 10 de un total de 66 registros

Ilustración 44: Pestaña Partes mecánicas SRVSA

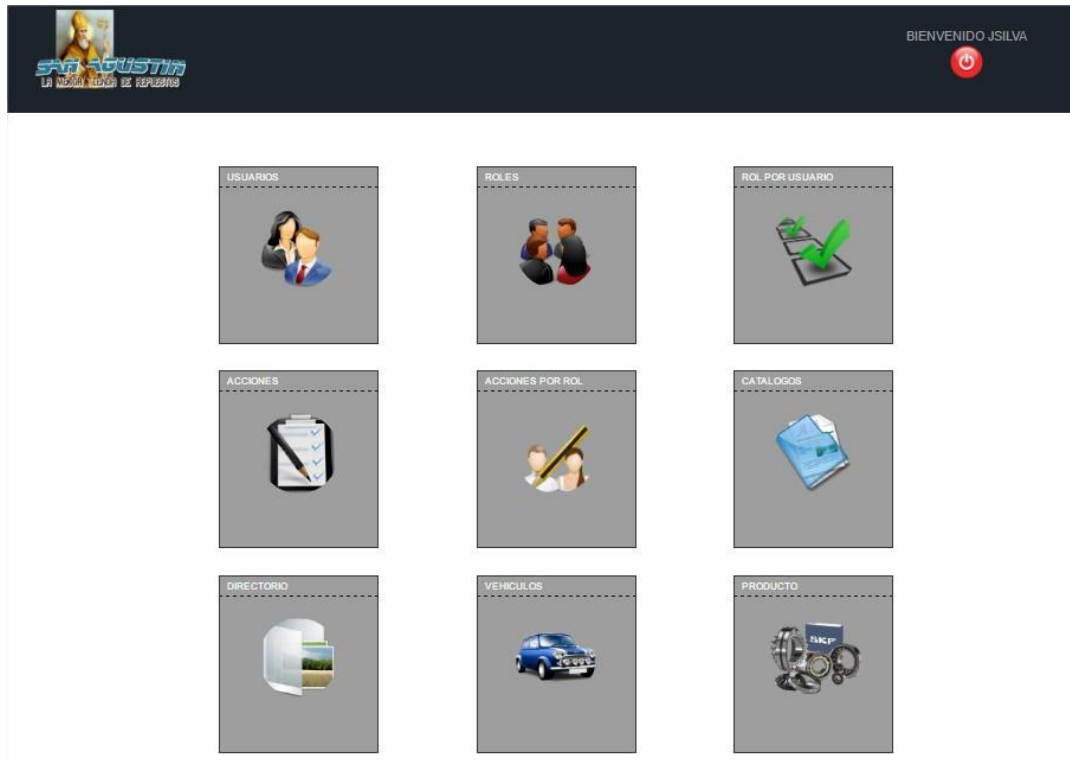


Ilustración 45: Pestaña Administración Superior SRVSA

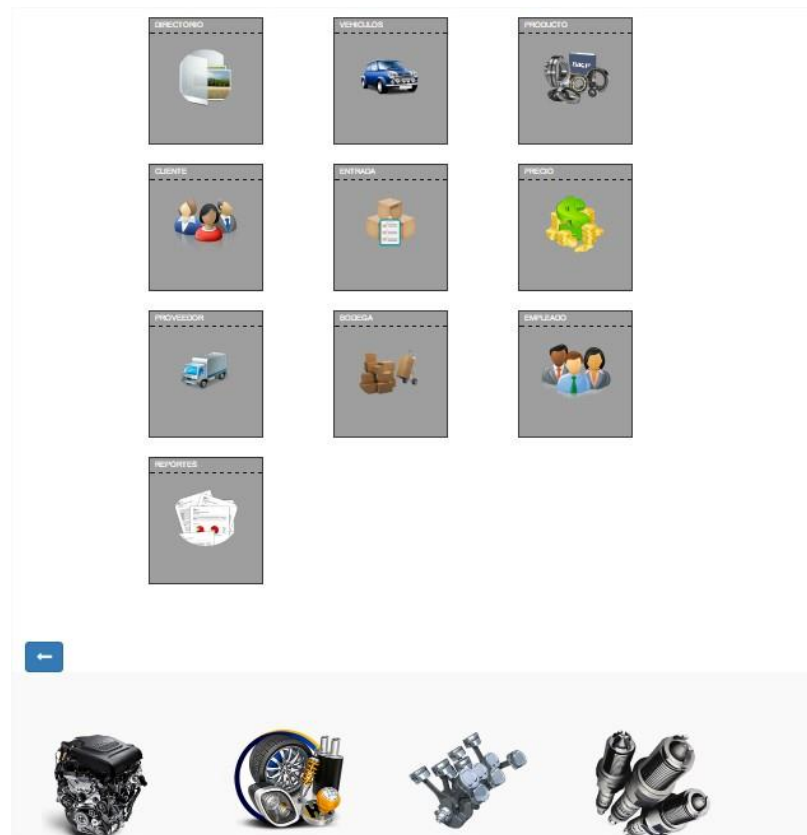


Ilustración 46: Pestaña Administración Interior SRVSA



Diseño de formularios de administración

Ingreso de vehículos

Agregar Vehículo

Marca: - Seleccione una marca - *

Modelo: - Seleccione un modelo - *

Version: - Seleccione una version - *

Tipo Combustion: - Seleccione el tipo de co - *

Cilindraje: *

Capacidad: 0.00 *

Año: 0 *

Guardar Cerrar

Marca	Modelo	Año	Editar
HYUNDAI	EXCEL	1989	[Editar]
HYUNDAI	EXCEL	1989	[Editar]
HYUNDAI	SONATA	1989	[Editar]
HYUNDAI	EXCEL	1990	[Editar]
HYUNDAI	EXCEL	1990	[Editar]
HYUNDAI	SONATA	1990	[Editar]

Ilustración 47: Interfaz de agregar vehículo

Editar Vehículo

Marca: HYUNDAI

Modelo: SONATA

Version: L

Tipo Combustion: GASOLINA

Cilindraje: L4 *

Capacidad: 1975.00 *

Año: 2015 *

Activo: ☒

Guardar Cerrar

Marca	Modelo	Año	Editar
HYUNDAI	SONATA	2015	[Editar]
HYUNDAI	SONATA	2015	[Editar]
HYUNDAI	SONATA	2015	[Editar]
HYUNDAI	SONATA	2015	[Editar]
HYUNDAI	TUCSON	2015	[Editar]
HYUNDAI	TUCSON	2015	[Editar]
HYUNDAI	TUCSON	2015	[Editar]
HYUNDAI	VELOSTER	2015	[Editar]
HYUNDAI	VELOSTER	2015	[Editar]
HYUNDAI	VELOSTER	2015	[Editar]

Mostrando registros del 481 al 490 de un total de 490 registros

Ilustración 48: Interfaz de editar vehículo



Ingreso de productos

Agregar Producto

Nombre

Descripcion

Codigo OEM Sub Codigo OEM

Peso Codigo Barra

Producto Padre Tipo Sistema

Marca Producto Precio

Cantidad Minima Cantidad Maxima

Vehiculo Compatibles

Ilustración 49: Interfaz de agregar producto



Editar Producto

Nombre: TENSOR

Descripción: TENSOR

Código OEM: 2441028000 Sub Código OEM:

Peso: Código Barra:

Producto Padre: - Seleccione el Producto padre - Tipo Sistema: MOTOR

Marca Producto: - Seleccione la marca del - Precio: - Seleccione el precio del -

Cantidad Mínima: Cantidad Máxima:

Vehículo Compatibles: *HYUNDAI EXCEL 1989 GL

Activo: ☒

[Guardar](#) [Cerrar](#)

Ilustración 50: Interfaz de editar producto

Diseño de formularios de administración de cotización.

Logo: SAN AGUSTIN LA MANA TIENDA DE REPUESTOS

BIENVENIDO JSILVA

INICIO PARTES OEM PARTES MECANICAS ADMINISTRACION COTIZACION

Código Oem: Buscar:

Cotización

Buscar:

No. Cotizacion	Cliente	Cerrada	Fecha	Editar	Cerrar	Imprimir
12	CONDOMINIO CENTRO COMERCIAL MANAGUA	<input checked="" type="checkbox"/>	24/01/2017	Editar	Cerrar	Imprimir
14	GRUPO SOL SA J0210000135547	<input checked="" type="checkbox"/>	02/02/2017	Editar	Cerrar	Imprimir
19	POLARIS ENERGY NICARAGUA,S.A.	<input type="checkbox"/>		Editar	Cerrar	Imprimir

Mostrando registros del 1 al 3 de un total de 3 registros

Primero Anterior 1 Siguiente Último

Ilustración 51: Interfaz de edición de cotización



1 de 1 100% Buscar | Siguiente

 **DE LA POLICIA NACIONAL**
2C AL NORTE 1/2 C. AL OESTE

Cotización

Nombre: LUIS FAVIO LOPEZ Fecha: 2/13/2017
Telefono: Asesor: JOSELINE SILVA

Cantidad	Descripción	Precio por unidad	Precio Total
1	ALTERNADOR	14.58	14.58
2	RODAMIENTO DELANTERO	14.58	29.16
5	RODAMIENTO DELANTERO	14.58	72.9
8	RODAMIENTO DELANTERO	14.58	116.64
3	CASE DELANTERO		

Ilustración 52: Interfaz de Reporte de Cotización

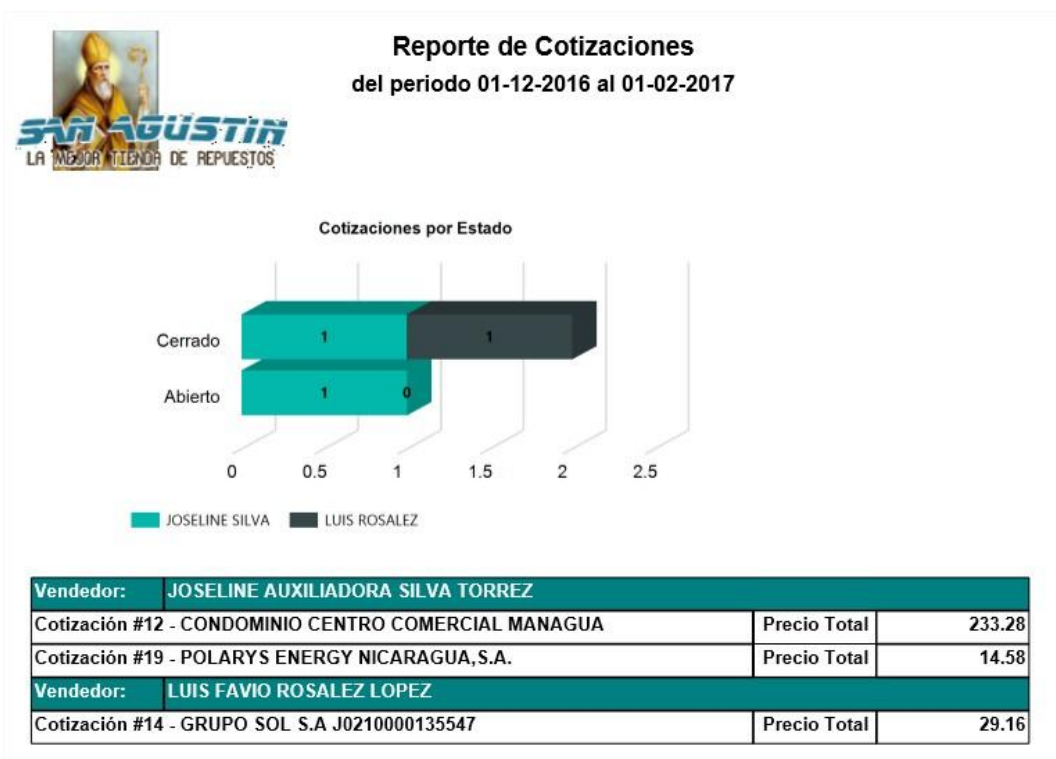


Ilustración 53: Interfaz de Reporte de Cotizaciones por vendedor



Conclusiones y Recomendaciones

En este capítulo presentamos los principales logros y limitaciones durante el desarrollo de la tesis monográfica, las experiencias adquiridas, las recomendaciones y conclusiones finales.

Limitaciones

Durante la exploración de herramientas a utilizar se hizo uso de **Unity3D**, **Unreal Development Kit**, **Torque 3D**, **CryEngine** entre otros, los cuales presentaron inconvenientes a la hora de diseñar una interfaz de usuario amigable, debido a que están orientados al desarrollo de videojuego, por lo tanto, la creación y manipulación de las tablas no era posible.

Debido a estas limitaciones se migro de plataforma utilizando como IDE desarrollo visual Studio con el patrón de desarrollo web MVC, en el cual gracias a una librería de JavaScript denominada **three.js** se pudo combinar los modelos 3D con el prototipo de sistema sin afectar el renderizado de los objetos 3D.

Experiencias y Logros

Dentro de los logros obtenidos y las experiencias adquiridas durante la realización de la tesis se destaca el aprendizaje y puesta en práctica de nuevos conocimientos, de los cuales se puntualizan:

- La metodología para el desarrollo rápido de aplicaciones RAD (acrónimo en inglés de rapid application development).
- La herramienta de Mapeo objeto-relacional para la plataforma MVC, como lo es Entity Framework.
- La implementación de Bootstrap, un framework para desarrollar interfaces y diseños web adaptables basados en HTML5 y CSS3.
- El aprovechamiento de técnicas de diseños para objetos 3D
- Compaginar objetos 3D con interfaces web a través de librerías de JavaScript



Conclusiones

La presente monografía constituye una solución a los problemas detectados en los procesos de venta de repuestos en la Empresa “**Repuestos San Agustín**”. Las principales conclusiones que se obtienen del trabajo de tesis son las siguientes:

El conocer las reglas del negocio fue fundamental para identificar y especificar los requerimientos de los usuarios de “**Repuestos San Agustín**”, lo que nos permitió el desarrollo óptimo y claro de un prototipo de software de acuerdo a las necesidades identificadas, logrando cumplir exitosamente con los objetivos inicialmente planteados.

La metodología RAD influyó positivamente en el desarrollo del proyecto, sirviendo como guía en la realización de todos los diagramas necesarios durante el proceso de análisis de requerimientos y de diseño, los cuales permitieron el desarrollo de un prototipo de software parametrizable, fácil de usar, adaptable a cambios y robusto.

El desarrollo del prototipo de Sistema Web de realidad virtual para la caracterización de piezas automotrices (SRVSA) es de utilidad para la coordinación de ventas de repuestos automotrices y a través de éste se ha logrado optimizar las ventas para brindar una mejor atención a sus clientes; al contar con una herramienta que les permita identificar las piezas automotrices de los vehículos, así como otras características útiles de conocer para decidir sobre la adquisición o no de la pieza de refacción.

Los medios, métodos, herramientas y técnicas de gestión del proyecto, de análisis y diseño del sistema, de codificación y pruebas, así como de despliegue e inducción del sistema, han sido una gran experiencia, que consideramos nos ha permitido fortalecer ampliamente nuestra formación académica y profesional.

Sin esta experiencia, no estaría completa nuestra formación como Ingenieros en Computación.

Esperamos que la realización de la presente tesis sirva como referencia a estudiantes que realicen en el futuro proyectos afines a este. Y para nosotros, esperamos que sea la motivación para seguir ampliando nuestros conocimientos y experiencias profesionales.



Recomendaciones

Recomendamos, dar continuidad al proyecto con el diseño 3D de otras piezas vehiculares, de las que la empresa tenga mayor demanda, para ampliar el catálogo de piezas 3D, haciéndolo progresivamente. Asimismo, el desarrollo de un módulo de facturación para el desarrollo de un sistema más completo que cumpla con las necesidades de la empresa.

Durante el levantamiento de requerimientos para la generación de cotizaciones, el personal de Repuestos San Agustín fue planteando otras funcionalidades propias de un sistema administrativo, que no se realizaron porque estaban fuera del marco de nuestro de trabajo. Se sugieren estas posibles mejoras que se pueden aplicar en un futuro al prototipo de sistema para mejorar sus funcionalidades:

- Implementar el módulo de ventas para completar el ciclo de facturación, que inicia con la búsqueda e identificación de piezas caracterizadas en el prototipo.
- Implementar un quiosco virtual, que permita a los clientes de la empresa buscar las piezas que desean cotizar por su propia cuenta, a fin de generar una proforma sin necesidad de la intervención de un vendedor.
- Envío de alertas de bajas de inventario por medio de correo electrónico.
- Desarrollar una aplicación móvil para futura ampliación, que permita a los vendedores de ruta mostrar los catálogos 3D de forma remota.



Bibliografía Web

- (s.f.). Obtenido de
http://sisbib.unmsm.edu.pe/bibvirtual/publicaciones/indata/v02_n1/produccion.htm
- (10 de 2008). Obtenido de <http://es.scribd.com/doc/7411856/Caracteristicas-de-C>
- (2016). Obtenido de Microsoft.com: <https://msdn.microsoft.com/es-es/library/bb630404.aspx>
- Autores, T. (s.f.). *Wikipedia*. Obtenido de <https://es.wikipedia.org/wiki/Three.js>
- Brezo, M. P. (2012). *Lancetalent*. Obtenido de <http://www.lancetalent.com/blog/6-buenos-motivos-para-trabajar-con-php/>
- Cabanes, N. (2012). *Introducción a la programación con C#*.
- DISTANCIA, U. N. (s.f.). Lección 5: Investigación pura, investigación Aplicada, Investigación profesional.
- E. Kendall, K. y. (2005). *Análisis y diseño de sistemas*. (Sexta Edición. ed.). México: Pearson Educación.
- EcuRed*. (s.f.). Obtenido de <http://www.ecured.cu/index.php/SQLite>
- Gálvez, M. A. (2004). Obtenido de
<http://www.tdx.cat/bitstream/handle/10803/5434/amgm1de1.pdf;jsessionid=EFD230AF49932EA3ECC76F4AEF2A26C2.tdx1?sequence=1>
- Grupo PSA Peugeot Citroën*. (s.f.). Obtenido de <http://www.psa-peugeot-citroen.com/fr/users/hbc00/web/html/htdocs/es>
- <http://www.swingalia.com/>. (2008). Obtenido de <http://www.swingalia.com/disenio/que-es-el-diseno-automotriz.php>
- Investigación, T. d. (2015).
- Kendall, K. E. (2011). *Análisis y diseño de sistemas* (8va edición ed.). Pearson Educación.
- Kennedy Bill, C. M. (2006). *HTML & XHTML: The Definitive Guide*, O'Reilly, 2006.
- Leon Shklar, R. R. (2003). *Web application architecture: principles, protocols, and practices*. John Wiley & Sons, Ltd.
- Libros Web*. (s.f.). Obtenido de Libros Web: https://librosweb.es/libro/ajax/capitulo_1.html
- LTDA, C. R. (s.f.). *CIPER Repuestos LTDA*. Obtenido de <http://www.ciper.cl/>



- Martel, M. a. (2002.). *Extreme Programming: Rapid Development for Web-Based Applications IEEE Internet Computing*.
- Martin, J. (1990). *Rapid Application Development*. MacMillan Publishing Co., ed.
- Martínez, F. J. (2011). *academia.edu*. Obtenido de http://www.academia.edu/5081919/Francisco_Javier_P%C3%A9rez_Mart%C3%ADnez_Presente_y_Futuro_de_la
- Matsuba, S., & Roehl, B. (1996). *Using VRML*. Indianapolis, IN: Que, ©1996.
- Microsoft. (s.f.). Obtenido de <https://msdn.microsoft.com/es-es/library/bb545450.aspx>
- Muñoz, D. F. (mayo de 2013). *Revista Digital Universitaria*. Obtenido de <http://www.revista.unam.mx/vol.14/num5/art06/index.html>
- Pollock, J. (2010). *JavaScript A Beginner's Guide*. McGrawHill Companies.
- Ramos, M. J. (s.f.). *Sistemas Gestores de Bases de Datos (Ciclo Formativo de Grado Superior)*. McGrawHill.
- Roberto Pérez Rodríguez, A. M. (s.f.). Obtenido de http://www.copimerainternacional.org/publicaciones/revista_techno/ci_tec/4_sistemas.pdf
- Seco, J. A. (s.f.). *El lenguaje de programación C#*.
- SIEMENS. (s.f.). Obtenido de <http://www.industry.siemens.com/verticals/global/es/automotive/manufacturing/process/body-in-white/pages/default.aspx>
- Technologies, U. (2014). Obtenido de <http://docs.unity3d.com/Manual/index.html>
- Thomas Powel, F. S. (2004). *JavaScript: The Complete Reference*, McGraw-Hill/Osborne.
- Unity3d. (s.f.). Obtenido de <http://docs.unity3d.com/Manual/UnityOverview.html>, Documentación en línea de Unity
- Vistosi, M. (2011). *TECNOLOGIA DE LAS COMPUTADORAS*. Obtenido de <http://vistosiblog.blogspot.com/2011/07/diccionario-informatico.html>
- WIKIPEDIA. (s.f.). Obtenido de https://es.wikipedia.org/wiki/Microsoft_SQL_Server



Anexos

Anexo 1: Plan de Implantación

El plan de implementación constituye una guía para apoyar el proceso de puesta en marcha del **Prototipo de Sistema de Realidad Virtual para la Caracterización de Piezas Automotrices (SRVSA)**, en el cuál se establecen los pasos a seguir y demás aspectos que se deben contemplar a lo largo de este.

El plan se divide en tres grandes áreas, que son la planificación, la organización y el control, donde se describen desde las actividades que se deben realizar, hasta el personal necesario para llevarlas a cabo, así como los controles necesarios para verificar el buen desarrollo de todo el proceso, con el fin de poder obtener los resultados deseados.

Objetivos del plan de implantación

Objetivo General

Establecer un plan que permita llevar a cabo la implementación del **Prototipo de Sistema de Realidad Virtual para la Caracterización de Piezas Automotrices**, tomando en cuenta los aspectos de planeación, organización y control.

Objetivos Específicos

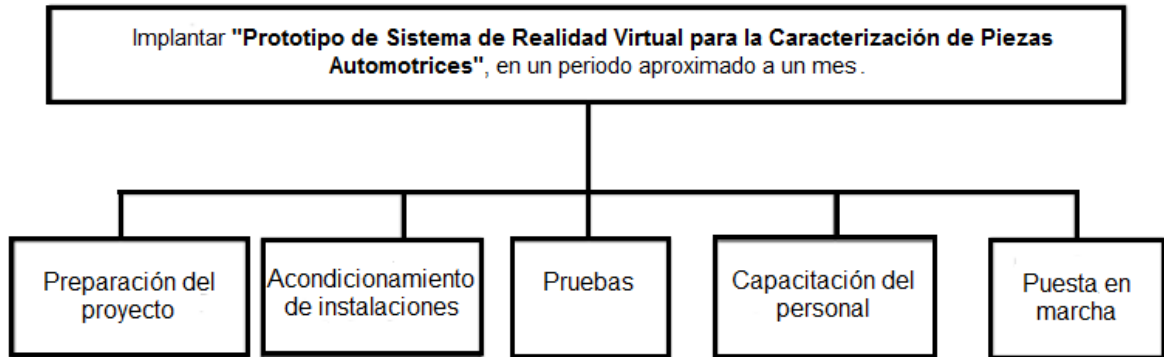
1. Realizar la planeación del plan de implementación, determinando las actividades a realizar y los recursos que serán asociados a éstas.
2. Elaborar la programación, asignando los recursos por actividad y fechas.
3. Definir la estructura organizativa del comité encargado de ejecutar el proyecto, estableciendo las funciones y responsabilidades de los miembros.

Planeación

Esta primera parte, contempla el desglose analítico; las cargas de trabajo (compuesto por el listado de actividades a realizar) y la programación de actividades (asignación de tiempos y recursos humanos para cada una de las actividades planteadas).



Diagrama de desglose analítico



Descripción de subsistemas

Para alcanzar el objetivo del proyecto, es necesario determinar procesos que nos permitan lograrlo. Dichos procesos son subsistemas que se desglosan a continuación:

Preparación del proyecto

Para la preparación del proyecto se realizarán las siguientes actividades de acuerdo al ámbito para la implementación respectiva:

1. Asignación del director del proyecto: Actividad en la que se decide quién será el encargado de dirigir la implementación del sistema.
2. Asignación del administrador de la red: Actividad en la que se selecciona a la persona que estará a cargo de la red que se implementará en la institución.
3. Asignación del jefe de ejecución: Persona encargada del plan de implementación, supervisando las actividades que permitan la coordinación del personal de repuestos san Agustín con el equipo de implementación.

Acondicionamiento de instalaciones

Para poder acondicionar el espacio físico, de tal forma que se instale el "SRVSA" dentro de la institución, se llevarán a cabo las siguientes actividades:

a) Acondicionamiento del espacio físico dentro del área de la institución

1. Evaluar las condiciones del espacio físico del local: Actividad en la que se determinan las condiciones evaluando la distribución del mobiliario existente y el espacio utilizado para el equipo informático.
2. Diseñar la distribución del equipo informático dentro del local: equipos como switches, cableado, impresoras, equipos de cómputo, mobiliario, etc.
3. Adecuación del local:



Se deben verificar los siguientes factores:

- a. Existencia de aire acondicionado en las áreas de vitales como sala de servidores.
- b. Infraestructura de red que permita la conexión estable de los equipos.
- c. Infraestructura física que ofrezca seguridad al equipo contra situaciones climatológicas u otras que afecten el buen funcionamiento de este.

b) Instalación del Software

1. Configuración de servidor de aplicaciones
2. Instalación de la base de datos de SRVSA en el servidor asignado a la aplicación.
3. Publicación de la aplicación en el IIS configurado en el servidor de aplicación.
4. Configuración de los accesos a la aplicación, así como los permisos de usuarios.

Pruebas

Para la realización de pruebas piloto del SRVSA, se desarrollarán las actividades siguientes:

- a) **Diseño de las Pruebas** Diseño de pruebas piloto que permitan detectar errores en la instalación de la aplicación informática y dificultades en la transmisión de datos por la red.
- b) **Realización de las Pruebas** Realización de las pruebas que permitan tener un software correctamente instalado.

Dentro de las pruebas el jefe de ejecución debe verificar los siguientes factores:

1. Comunicación de las estaciones de trabajo con el servidor.
 2. Conexión de SRVSA en las estaciones de trabajo con la base de datos en el servidor.
 3. Comunicación de las estaciones de trabajo con las impresoras.
- c) **Análisis de los resultados de las Pruebas** Verificar los resultados obtenidos en las pruebas efectuadas y realizar correcciones.

Capacitación del personal

Para la capacitación del personal de Repuestos San Agustín, se llevarán a cabo las siguientes actividades:

a. Preparación de la capacitación

- a. Diseño de la capacitación para los distintos niveles de usuarios.
- b. Preparación de todo el material que será utilizado en las capacitaciones, esto incluirá los manuales de usuario.
- c. Preparación de los grupos a capacitar dividiendo al personal en grupos de acuerdo a los niveles de acceso; para que la capacitación sea impartida de acuerdo a lo que cada usuario necesita.

b. Capacitación del Personal

Esta actividad permitirá el adiestramiento del personal que labora en Repuestos San Agustín en las opciones a las que tendrán acceso dentro del sistema.



Puesta en marcha

Para la puesta en marcha de SRVSA se realizará la conversión del sistema manual actual, desarrollando las siguientes actividades:

1. Operación paralela del sistema de venta de Repuestos San Agustín con el SRVSA, con el objetivo de verificar que los datos generados (Códigos OEM, precios, existencias) por el sistema sean correctos, y que los usuarios se familiaricen gradualmente con este.
3. Arranque del SRVSA quedará operando una vez se haya comparado en un tiempo de 15 días con el sistema manual actual, y todos los errores encontrados hayan sido corregidos a través de la comparación hecha.

Programación para la Implantación

Cronograma de actividades

Actividad	Duración días hábiles	Fecha inicial	Fecha final
Preparación del proyecto	1		
Asignación del director del proyecto	1	20/01/2017	20/01/2017
Asignación del administrador de la red			
Asignación del jefe de ejecución			
Acondicionamiento de instalaciones	11		
Evaluar las condiciones del espacio físico del local	1	23/01/2017	23/01/2017
Diseñar la distribución de los equipos informáticos dentro del local	1	24/01/2017	24/01/2017
Adecuación del local	5	25/01/2017	31/01/2017
Configuración del servidor de aplicaciones	1	01/02/2017	01/02/2017
Instalación de la base de datos en la aplicación	1	02/02/2017	02/02/2017
Publicación de la aplicación en el IIS	1	03/02/2017	03/02/2017
Configuración del acceso al servidor	1	04/02/2017	04/02/2017
Pruebas	5		
Diseño de las pruebas	1	06/02/2017	06/02/2017
Realización de las pruebas	2	07/02/2017	08/02/2017
Análisis de los resultados	2	09/02/2017	10/02/2017
Capacitación del personal	10		
Diseño de la capacitación para los distintos niveles de usuario	3	11/02/2017	14/02/2017
Preparación del Material de capacitación	1	15/02/2017	15/02/2017
Preparación de los grupos a capacitar	1	16/02/2017	16/02/2017
Capacitación del personal	5	17/02/2017	22/02/2017
Puesta en marcha	10		
Operación en paralelo del sistema	6	23/02/2017	02/03/2017
Análisis de resultados	2	03/03/2017	04/03/2017



Arranque del sistema	2	06/03/2017	07/03/2017
Total	37		

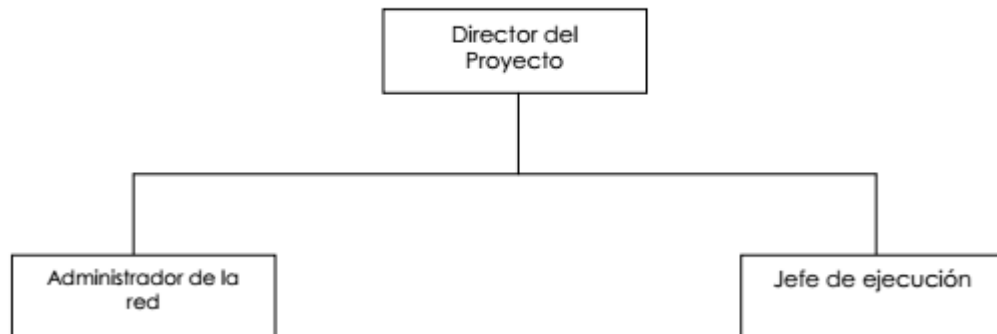
Organización

La organización, comprende el establecimiento del marco organizativo sobre el que funcionará la ejecución del proyecto; para lo cual se plantea la estructura organizativa de la unidad ejecutora del proyecto, sus funciones, y una matriz de responsabilidades.

Estructura organizativa de la unidad ejecutora

Para la implantación de este sistema se necesitan definir las actividades en el área de instalación de los componentes de software, pruebas de implementación y las capacitaciones al personal; estableciéndose los responsables de dichas actividades, por lo que el primer paso será establecer la estructura organizativa, la cual está constituida de la manera siguiente:

Organigrama de Unidad Ejecutora



La implementación estará a cargo de un Director de Proyecto, el cual será el responsable de dirigir todo el proceso y se basará en lo definido en el presente plan. Este tendrá a su cargo al administrador de la red y la persona encargada de la ejecución del proyecto, así como también supervisará la parte de instalación de la red.



Anexo 2: Evaluación Presupuestal

Factibilidad Técnica

Inventario

Hardware

Repuestos San Agustín

5 computadoras Dell OptiPlex 3040 Micro

- **Sistema Operativo**
Windows 8.1 64bits
- **Procesador**
Intel Core i3-6100T Dual Core, 3MB, 4T, 3.2GHz, 35W
- **Memoria RAM**
4GB. 1600MHz DDR3
- **Disco duro**
500 GB
7200 rpm
- **Unidad de DVD**
CD-ROM/DVD
- **Gráficos:**
Intel HD (hasta 731 MB)
- **Puertos:**
1 para audífono
1 LAN
4 USB 3.0
2 USB 2.0
3 conectores de audio
1 HDMI



1 VGA

Interfaz de Red:

Interfaz de red Ethernet 10/100BT integrada.

Tecnologías inalámbricas:

LAN inalámbrica 802.11b/g/n.

1 impresora hp deskjet 1000

Resolución de impresión: Negro (óptima): Hasta 600 ppp; Color (óptimo): Hasta 4800 x 1200 ppp optimizados

1 escáner hp scanjet 3770

Dimensiones de 287 x 450 x 65 milímetros.

Peso de 3 kg.

Resolución óptica máxima de 1200 x 2400 ppp.

Tamaño de digitalización es de hasta 216 x 297.

Incluye una fuente de luz CCF y un CDD que le otorga la capacidad de alcanzar una profundidad de color de 48 bits.

Interfaz es USB 2.0

Software

El Sistema de Ventas San Agustín está desarrollado en Access, se encuentra instalado en un equipo que es usado de servidor central para el sistema, los equipos que hacen uso del sistema de ventas requieren tener el acceso directo con la ruta del servidor para hacer uso del sistema, si hay problemas de red el acceso directo debe ser configurado nuevamente.



Solución Técnica propuesta

Sistema web bajo el patrón de arquitectura MVC que permita a múltiples usuarios acceder a la base de datos del sistema al mismo tiempo con tiempos de respuesta bajos.

Estrategia del Hardware/Comunicaciones

Es recomendable un servidor HP 500B Micro torre que se ubique en una oficina debidamente climatizada,

Este servidor deberá contener el gestor de base de datos SQL Server 2014. Además, deberá contar con IIS habilitado para el correcto funcionamiento de sistema web.

En este servidor se almacenará la base de datos del sistema, la instalación y configuración de firewalls para la protección de la información. El envío y recepción de información de las estaciones clientes al servidor bajo el protocolo http de autenticación encriptado, el cual estará configurado para que las únicas personas que puedan acceder a la información sean quienes tienen una cuenta de usuario.

Al servidor se conectarán todos los equipos con los que cuentan la empresa, los cuales estarán ubicadas 3 en área de ventas, una en caja y otra en el área de despacho.

Los equipos adicionales que se conectarán a la red serán las computadoras portátiles y teléfonos personales de los gerentes.

Para el acceso a la red 2 de las estaciones de trabajo ubicadas una en despacho y otra en área de ventas deberán ser cableadas. Es necesario realizar la adquisición de un switch de 8 puertos para este fin

Estrategia del Software

Se instalará en todos los equipos de desarrollo en software Windows 10, versión más compatible con el IDE Visual estudio 2015 para el desarrollo de la. Como gestor de base de datos se usará el software SQL Server 2014 el cual estará instalado en el servidor para el acceso a la base de datos desde las estaciones clientes.

Estrategia de RRHH

Para la puesta en marcha del proyecto se necesitan:

Un Analista de Sistemas: Experiencia mínima de 2 años en el análisis de sistemas, con capacidades de líder y de negociación, conocimiento y experiencia de programación en



los lenguajes HTML, JavaScript, JQuery, Lenguaje de programación C#, patrón de diseño MVC 4 y SQL server 2014, con grandes valores éticos y morales.

Un programador: experiencia mínima de dos años en el desarrollo de software, con conocimientos en HTML, JavaScript, JQuery, Lenguaje de programación C#, patrón de diseño MVC 4 y SQL server 2014, que sea capaz de trabajar bajo presión.

Un Diseñador Gráfico: experiencia mínima de dos años en uso de herramientas de modelaje 3D (3ds Max, Blende, Maya), usos de herramientas de diseño adobe CSS 5.

Múltiples Usuarios.

Un implantador del sistema: experiencia mínima de dos años.

Metodología de Desarrollo

Desarrollo Rápido de Aplicaciones

Estrategia de Procesamiento

Los equipos de desarrollo del sistema estarán distribuidos de forma tal que satisfagan las necesidades de todo el personal involucrado en el proyecto; es decir el programador, el analista de sistema, el diseñador gráfico tendrá a su cargo una computadora por persona para optimizar sus condiciones de trabajo. El implantador del sistema será quien manipule y configure el servidor y la red; se encargará instalar la aplicación desarrollada en el servidor, así como la base de datos y garantizar que todas las estaciones clientes tiene acceso a la aplicación web en el servidor.

Costos

Costos de Inversión

HARDWARE			
Cantidad	Descripción	Precio Unitario	Precio Total
4	UPS Forza NT 5010 para protección de equipos	\$ 40	\$ 160
1	UPS Forza FX-900LCD de respaldo para el Servidor	\$140	\$ 140



1	Servidor HP 500B Micro torre	\$ 658	\$ 658
Subtotal	\$ 958		
SOFTWARE			
Cantidad	Descripción	Precio Unitario	Precio Total
1	Licencia para el gestor de base de datos	Gratuita	U\$ 0
4	Licencia para software de desarrollo	Gratuito	U\$ 0
1	Compra de licencias del sistema operativo	U\$ 619	U\$ 619
Subtotal	U\$ 619		
COMUNICACIONES			
Cantidad	Descripción	Precio Unitario	Precio Total
25 mts.	Cables de red	U\$ 2	U\$ 50
1	Swicth NEXXT NW223NXTXX 8 puertos	U\$ 47.5	U\$ 47.5
Subtotal	U\$ 97.5		
TOTAL			

Costos de Desarrollo

Cantidad	Rol	Salario Mensual	Meses	Total
1	Analista de Sistemas	U\$ 700	3	U\$ 2,100
1	Programador	U\$ 450	3	U\$ 3,350
1	Diseñador Grafico	U\$ 400	2	U\$ 1,200
1	Implantador de Sistemas	U\$ 600	1	U\$ 600



Cantidad	Rol	Salario Mensual	Meses	Total
TOTAL	U\$ 7,250			

Costos Complementarios

Cantidad	Descripción	Precio Unitario	Precio Total
1	Mantenimiento correctivo	U\$ 35	U\$ 35
1	Técnicos para capacitación	U\$200	U\$200
TOTAL	U\$ 235		

Costo Total

Cantidad	Totales
Costo de Inversión	U\$ 1,674.5
Costo de Desarrollo	U\$ 7,250.00
Costos Complementarios	U\$ 235
GRANT TOTAL	U\$ 22,939.86